Disciplined Machine Learning

DRAFT
May 18, 2026

Disciplined Machine Learning

Hao Zhu

Department of Computer Science
University of Freiburg

Joschka Boedecker

Department of Computer Science
University of Freiburg

Contents

Preface

This book is about *disciplined machine learning*, which aims at providing a systematic and principled way to design and analyze machine learning models, especially those where domain knowledge and model structures are (or should be) explicitly incorporated.

Motivation

Our initial motivation for writing this book was driven by two (perhaps not so recent) developments in computational mathematics and optimization.

The first is *convex optimization*, which has been developed and studied for several centuries, but it was not until the 1990s that people started to realize that it is far more prevalent across domains than previously thought. Today, convex optimization has become a mature technology that is widely used in science and engineering. Roughly speaking (and perhaps optimistically), if a problem (beyond least squares and linear programming) can be formulated as a convex optimization problem, then it is essentially solved.

The second development is *disciplined optimization*, a systematic way to construct and analyze optimization problems so that a human or a computer program can automatically determine a problem's structure and properties and select appropriate numerical algorithms to solve it. In practice, disciplined optimization allows practitioners to focus on *defining* the problem according to application needs, while leaving the details of *solving* the problem to the computer. This is close to the idea of *"say what you want, not how to do it."*

Disciplined optimization was initially designed for convex optimization in the early 2000s, but it has also been extended to some nonconvex problems over the last decade. As a generic framework for mathematical optimization, it often works quite well across a wide range of problems.

With these advances, we believe that disciplined optimization and analysis is something that everyone interested in machine learning should be familiar with, at least a little bit. Conceptually, a disciplined view of machine learning models helps practitioners form an overall picture and understand relationships between different models. Practically, it helps identify essential *building blocks* that recur across many models, making it easier to reuse them when designing more complex ones. This, in turn, enables people to *customize* and *prototype* machine learning models quickly and systematically for specific applications. We develop this perspective throughout the book, and we hope readers will find it useful in their own work.

Goal of this book

There are several languages for representing machine learning models. Traditionally, people have been using natural language and mathematical notation to describe the models, and this is still the most common way to do it. Some models are better known in the form of an algorithm or pseudocode, corresponding to a specific procedure for model parameter estimation or inference.

Disciplined machine learning, instead, uses the language of optimization to represent the models. In this context, different models are characterized by their *inverse problems*, which are the optimization problems to be solved for estimating the model parameters from data. This representation provides a higher-level description of machine learning models, with the advantage of allowing us, *e.g.*, to quickly estimate the effort required to fit and use a model and, when prior knowledge about model structure is available, to adapt the model efficiently and systematically. Of course, some models are well known or can be easily represented in terms of inverse problems (*e.g.*, linear regression), while doing so for others (including many very famous ones, *e.g.*, principal component analysis) is not straightforward, or less known.

> *The main goal of this book is to help readers develop the disciplined mindset, knowledge, and skills to formulate, analyze, and customize machine learning models via inverse problems.*

As the first intuition suggests, developing a disciplined machine learning mindset can be mathematically demanding, especially for readers who are (or have been) primarily interested in using these models in various applications. We do, however, observe that those invested efforts are often worthwhile and usually pay off very well in the long run.

This focus differentiates this book from many other textbooks. There are several books on the theory and analysis of machine learning models, which are more focused on the mathematical properties of the models. Some books focus on the practical implementation of machine learning models, *i.e.*, the algorithms and software. Several other books provide highly practical and application-oriented introductions to machine learning, which are more focused on the real-world use of these ideas. This book is designed to lie somewhere in between, with the primary focus on the systematic design and high-level modeling of machine learning models, via the so-called *disciplined modules* that build up the inverse problems.

It is also worth noting what this book is *not* about. This book is not about state-of-the-art models and research in machine learning, and it is also not about specific applications of machine learning in real-world scenarios. Although we do describe some applications of various models on different tasks, the main purpose of these texts is to associate the abstract disciplined machine learning principles with concrete examples, and to show how the former can be used to design and analyze the latter. Furthermore, we make no attempt to cover the full set of specialized algorithms for solving an inverse problem corresponding to a highly customized model. Instead, we provide essential principles for recognizing the structure (such as convexity) of the problem, so that a potential user could make use of the generic disciplined optimization frameworks to implement a working solver. In our expe-

rience, this is usually effective for moderate-sized problems; when it fails, it still provides a good starting point for developing more advanced algorithms.

Audience

This book is intended for (under)graduate students, researchers, scientists, engineers, and practitioners in fields like machine learning and data mining. We would like to make the book accessible to a wide audience, and hence, aside from some fundamental concepts and tools, the material was selected to be broad and interesting enough for a wide range of readers. The trade-off is that we do not cover specific applications in depth, nor do we aim to cover the latest research in the field. We apologize to readers who are already experts in the related community, and we hope they can still find useful material in the book.

The background required of the reader is the knowledge of basic calculus, linear algebra, and probabilities. These are the only prerequisites that we use in the book without much explanation, and we assume the reader is familiar with them. If the reader has been exposed to mathematical analysis (*e.g.*, norms and topology) and optimization, he or she should be able to follow every statement and discussion in this book. None of these is essential, though; we provide the necessary reviews of the needed material from these areas in the text and appendices. With this, we hope that the content of the book should be understandable to a wide range of readers with undergraduate level of mathematical training, including those who are not specialized in optimization.

We also highly recommend the book *Convex Optimization* by Boyd and Vandenberghe [BV04] as a companion to this book. Their work includes almost all the necessary knowledge on convex optimization needed in practice, and is an excellent resource for further study. Some of the background material presented in this book is adapted from their work, and the reader will notice soon that our treatment of the notation and basic concepts about optimization is heavily influenced by their ideas and presentation.

Acknowledgments

We would like to thank many people who have contributed to the development of this book, including our colleagues and students at the University of Freiburg, and the many people who have provided feedback on the early drafts of the book, including those who attended our lectures on this material at the University of Freiburg and elsewhere.

We want to single out Stephen Boyd for special acknowledgment, who has inspired our initial interest in disciplined optimization. We have learned and emulated much of Boyd's style of extreme clarity in mathematical writing. Readers who are familiar with Boyd's work may recognize his influence.

Comments on the current version

Currently, we consider this book to be a draft. We will update it from time to time, adding more material and fixing typos as we become aware of them. So you may wish to periodically check whether a newer version is available.

We also welcome any comments and suggestions for improvement, including, *e.g.*, examples, exercises, and missing references to be added. If you find any errors in the current manuscript, please do let us know as well.

Hao Zhu *Freiburg, Germany*
Joschka Boedecker *May 18, 2026*

Chapter 1

Introduction

In this introduction, we provide an overview of mathematical modeling and machine learning, from the perspective of inverse problems and optimization, and focus on the case where prior information is available. We also briefly outline the structure of this book and setup our notation. The concepts introduced informally and statements made without argumentation will be treated with more detail in later chapters.

1.1 Modeling with prior information

1.1.1 Mathematical modeling

Machine learning is often concerned with building parameterized mathematical *models* that can learn from data and make predictions or decisions based on the learned knowledge. These models are typically represented as functions that map some *inputs* (or *features*) to *outputs* (or *responses*, *observations*), and are characterized by a set of *parameters* that determine their behavior.

This book characterizes machine learning models by their *fitting* or *inverse problems*, which involves estimating the model parameters from observed data. Roughly speaking, fitting a machine learning model to data consists in finding the parameters that minimize some certain *objective* (or *cost*, *loss*) function, which, for example, might be a measure of misfit or prediction error between the observed data and the values predicted by the model, or a statistical measure of the unlikeliness or implausibility of the parameter values.

Mathematically, fitting a machine learning model can be formulated as an *optimization problem* with the following structure:

$$\text{minimize} \quad f_0(x) \tag{1.1}$$

where $x = (x_1, \ldots, x_n) \in \mathbf{R}^n$ is the *optimization variable* (or just *variable*) that represents the model parameters, and $f_0 \colon \mathbf{R}^n \to \mathbf{R}$ is the *objective* function that measures the quality of the model fit. A point $x^\star \in \mathbf{R}^n$ is called a *solution* of (or *solves*) the problem (1.1) if it achieves the lowest objective value among all other

points, *i.e.*, satisfies $f(x^\star) \leq f(x)$ for all $x \in \mathbf{R}^n$. Solving the optimization problem (1.1) finds the model parameter values that give the smallest misfit or prediction error with the observed data, or the most plausible parameter values according to the chosen statistical measure.

1.1.2 Disciplined machine learning

It is often beneficial to incorporate prior knowledge and structural information about the model into the fitting process. For example, we may know that certain parameters should be nonnegative, or that the model should exhibit sparsity or smoothness. In such cases, the inverse problem (1.1) can be extended to include *constraints* and *regularization* terms, *i.e.*,

$$\begin{array}{ll} \text{minimize} & f_0(x) + \phi(x) \\ \text{subject to} & x \in C, \end{array} \tag{1.2}$$

where the additional term $\phi\colon \mathbf{R}^n \to \mathbf{R}$ is called the *regularization function*, or *regularizer*, and the set $C \subseteq \mathbf{R}^n$ is the *constraint set*. A point $x \in \mathbf{R}^n$ is said to be *feasible* to (1.2) if it satisfies $x \in C$. If the constraint set C is equal to the whole space $\mathbf{R}^n$, then there are effectively no constraints, and the problem (1.2) reduces to (1.1); if the constraint set C is empty, *i.e.*, there exists no point in C, then the problem (1.2) is said to be *infeasible*.

Compared to (1.1), the regularization function ϕ helps to promote certain desired properties in the solution (such as sparsity or smoothness), while the constraint set C restricts the feasible solutions to those that satisfy specific conditions (such as nonnegativity or definiteness). In this way, the fitting process can be made more robust and better aligned with the underlying characteristics of the model and the data (as already known or the user may assume or expect).

We call this systematic and principled way of formulating and analyzing machine learning models via their inverse problems, and, in particular, of incorporating prior information explicitly through regularization and constraints, *disciplined machine learning*.

Specify a constraint set

Technically, in practice, the constraint set C is often represented via the solution set of a system of inequalities and equalities:

$$C = \left\{ x \in \mathbf{R}^n \;\middle|\; \begin{array}{l} f_i(x) \leq 0,\ i = 1, \ldots, m \\ h_i(x) = 0,\ i = 1, \ldots, p \end{array} \right\},$$

where $f_i\colon \mathbf{R}^n \to \mathbf{R}$ are called *inequality constraint functions*, and $h_i\colon \mathbf{R}^n \to \mathbf{R}$ are called *equality constraint functions*. As a simple example, the nonnegativity constraint $x \in \mathbf{R}^n_+$ (where $\mathbf{R}^n_+$ is the *nonnegative orthant* in $\mathbf{R}^n$) can be represented via the inequality constraint $x \succeq 0$, where $\succeq$ denotes the *componentwise inequality*, *i.e.*, $x_i \geq 0$ for all $i = 1, \ldots, n$. Therefore, the inverse problem (1.2) is often written

as

$$\begin{array}{ll} \text{minimize} & f_0(x) + \phi(x) \\ \text{subject to} & f_i(x) \leq 0,\ i = 1, \ldots, m \\ & h_i(x) = 0,\ i = 1, \ldots, p. \end{array} \tag{1.3}$$

In order to distinguish between the two forms of inverse problems given by (1.2) and (1.3), the problem (1.2) is sometimes referred to as the *abstract form* inverse problem.

It is easily seen that the formulation (1.2) is more general than (1.3). However, it is often more convenient to work with the explicit constraint functions as in (1.3) (and it is indeed the form that most modern optimization problem solvers usually accept), since it is in general much easier to check if a given point is feasible by verifying if the inequalities and equalities are all satisfied, than evaluating if the point belongs to the abstract constraint set.

We will *not* use the abstract form (1.2) very often in this book, except for some conceptual discussions (such as those in the above paragraphs), but it is still useful to keep in mind its existence and generality.

1.1.3 Representation of prior information

According to the previous discussion on the inverse problem (1.2), we may notice that prior information about the model can be integrated into the inverse problem in two ways, *i.e.*, via the regularization function ϕ or via the constraints. The choice of a specific representation often depends on our belief on the faithfulness of the prior information and the specific modeling context: Constraints are often used to represent hard prior information that must be satisfied by the solution (*i.e.*, any violation of the constraints is unacceptable), while regularization terms are often used to represent soft prior information that we would like the solution to exhibit, but may not be strictly necessary.

For example, suppose we want to fit a model with parameter $x \in \mathbf{R}^n$, which is assumed to be sparse (*i.e.*, having many zero entries). Prior information on parameter sparsity can be represented using an ℓ_1-norm regularization term

$$\phi(x) = \lambda \|x\|_1 = \lambda \sum_{i=1}^{n} |x_i|,$$

where $\lambda > 0$ is the regularization coefficient that controls the strength of the sparsity promotion. By varying λ from small to large values, we can obtain solutions with different levels of sparsity (from dense to sparse), allowing us to explore the trade-off between model fit and parameter sparsity. (This idea is discussed more generally and in more detail in chapter 5.) In other words, the regularization function can be interpreted as a secondary objective that encourages certain properties in the solution, which we would like to achieve but are willing to compromise on if necessary (*e.g.*, to improve model fit). Alternatively, we can impose a constraint on the number of nonzero entries in x, such as $\mathbf{card}\, x \leq k$, where $\mathbf{card}\, x$ is the *cardinality* of the vector x, *i.e.*, the number of its nonzero entries. This constraint

directly restricts the solution (or actually, any candidate value of the model parameter x) to have at most k nonzero entries, enforcing a hard limit on the sparsity level.

Another more or less technical consideration regarding the regularization or constraint representation of prior information is the computational aspect: The choice of representation can have a significant influence on the complexity and solvability of the resulting optimization problem. Again, consider the sparsity prior information on the model parameter $x \in \mathbf{R}^n$: Using the ℓ_1-norm regularization $\phi(x) = \lambda\|x\|_1$ often leads to a convex optimization problem that can be efficiently solved, while imposing the cardinality constraint $\mathbf{card}\, x \leq k$ results in a combinatorial optimization problem that is generally very hard to solve exactly.

1.2 Some basic examples

In this section, we describe two very widely known and used special classes of machine learning models as some basic examples. We will revisit these models with much more detail in later chapters.

1.2.1 Least squares

A *least squares* problem corresponds to fitting a *linear model*, which has the form $a^T x$ (with a being the input feature), to data by minimizing the sum of squared differences between the observed outputs and the model predictions. Mathematically, given a dataset $\{(a_i, b_i)\}_{i=1}^m$, where $a_i \in \mathbf{R}^n$ are the input features and $b_i \in \mathbf{R}$ are the observed outputs, the least squares problem can be formulated as

$$\text{minimize} \quad f_0(x) = \|Ax - b\|_2^2 = \sum_{i=1}^m \left(a_i^T x - b_i\right)^2, \tag{1.4}$$

where the problem variable $x \in \mathbf{R}^n$ is the model parameter to be estimated. Here, $A \in \mathbf{R}^{m\times n}$ (with $m \geq n$), a_i^T are the rows of the matrix A, and $b = (b_1, \ldots, b_m) \in \mathbf{R}^m$. In statistics, the least squares problem (1.4) is sometimes called a *regression problem*, where $a_1, \ldots, a_m$ are the *regressors* and the solution is called the *regression of b onto the regressors*.

Least squares problem is one of the few inverse problems that has analytical solution, which is given by

$$x^\star = \left(A^T A\right)^{-1} A^T b$$

(provided the matrix A has full rank, *i.e.*, the columns of A are independent, $\mathbf{rank}\, A = n$). It can be shown that the $x^\star$ given above minimizes the objective of least squares, *i.e.*, satisfies

$$\|Ax^\star - b\|_2^2 \leq \|Ax - b\|_2^2$$

for all $x \in \mathbf{R}^n$, and therefore solves the problem (1.4).

Regularized least squares

There are several very famous variations of the least squares problem, by adding different regularization functions to the objective. The *Tikhonov regularization least squares*, which is also known as the *ridge regression*, adds an ℓ_2-norm squared regularizer to the least squares objective, resulting in the following inverse problem:

$$\text{minimize} \quad \|Ax - b\|_2^2 + \lambda \|x\|_2^2,$$

where $\lambda > 0$ is the regularization coefficient. The ridge regression problem also has an analytical solution, given by

$$x^\star = \left(A^T A + \lambda I\right)^{-1} A^T b.$$

Since $A^T A + \lambda I$ is always invertible for any $\lambda > 0$, ridge regression can be applied even when A is not full rank. The solution of ridge regression tends to be 'small' (*i.e.*, with many entries close to zero) due to the presence of the regularizer $\|\cdot\|_2^2$, and is therefore commonly used in practice to prevent overfitting and improve the stability of the solution, especially when the number of features n is large compared to the number of samples m.

Another similar variation is the *ℓ_1-regularized least squares*, which is also named *lasso regression*, corresponding to the following inverse problem:

$$\text{minimize} \quad \|Ax - b\|_2^2 + \lambda \|x\|_1,$$

where $\lambda > 0$ is the regularization coefficient. The ℓ_1-regularizer promotes sparsity (*i.e.*, with only a few nonzero entries) in the solution, leading to models that 'use' only a subset of the features, and hence, lasso regression is often used for feature selection and finding models that are easier to interpret. However, unlike least squares and ridge regression, there exists no analytical solution for lasso regression problems.

Constrained least squares

A (linear equality) *constrained least squares* has the form

$$\begin{array}{ll} \text{minimize} & \|Ax - b\|_2^2 \\ \text{subject to} & Cx = d, \end{array} \tag{1.5}$$

where $A \in \mathbf{R}^{m \times n}$, $C \in \mathbf{R}^{p \times n}$, $b \in \mathbf{R}^m$ and $d \in \mathbf{R}^p$ are given data. Assume C has full rank, it is only interesting when $Cx = d$ is an underdetermined system of linear equations, *i.e.*, $p < n$, since otherwise, if $Cx = d$ has a unique solution x_0, then x_0 is naturally the solution of (1.5); if $Cx = d$ has no solution, then (1.5) is infeasible. An important special case of (1.5) is when $A = I$ and $b = 0$:

$$\begin{array}{ll} \text{minimize} & \|x\|_2^2 \\ \text{subject to} & Cx = d, \end{array}$$

which is called a *least norm problem*, since it seeks the vector of smallest or least ℓ_2-norm that satisfies the linear equations $Cx = d$.

We can replace the equality constraint in (1.5) by an inequality constraint, resulting in the *inequality constrained least squares* problem:

$$\begin{array}{ll} \text{minimize} & \|Ax-b\|_2^2 \\ \text{subject to} & Cx \preceq d, \end{array}$$

where the system of linear inequalities $Cx \preceq d$ defines a polyhedron in $\mathbf{R}^n$, from which the solution could only be selected. As a special case, when $C = -I$ and $d = 0$, we have the *nonnegative least squares* problem:

$$\begin{array}{ll} \text{minimize} & \|Ax-b\|_2^2 \\ \text{subject to} & x \succeq 0, \end{array}$$

which requires that the solution x must be componentwise nonnegative.

1.2.2 Principal component analysis

Principal component analysis (PCA) is one of the oldest machine learning models and is widely used in data analysis for dimensionality reduction and feature extraction. PCA seeks to find the best rank-k approximation of a given data matrix $A \in \mathbf{R}^{m\times n}$ in the least squares sense, by solving the following inverse problem:

$$\begin{array}{ll} \text{minimize} & \|Z-A\|_F^2 \\ \text{subject to} & \mathbf{rank}\, Z \leq k, \end{array} \tag{1.6}$$

where the problem variable $Z \in \mathbf{R}^{m\times n}$ is the rank-k approximation of A, and $\|\cdot\|_F$ is the *Frobenius norm*, given by

$$\|U\|_F = \left(\mathbf{tr}(U^TU)\right)^{1/2} = \left(\sum_{i=1}^m \sum_{j=1}^n U_{ij}^2\right)^{1/2}.$$

The rank constraint can be encoded implicitly by decomposing Z as $Z = XY$, where $X \in \mathbf{R}^{m\times k}$ and $Y \in \mathbf{R}^{k\times n}$, leading to the equivalent formulation of PCA problem:

$$\text{minimize} \quad \|XY-A\|_F^2 \tag{1.7}$$

with variables $X \in \mathbf{R}^{m\times k}$ and $Y \in \mathbf{R}^{k\times n}$.

Compared to the inverse problem formulation (1.7), PCA is better known in the form of its analytical solution via *singular value decomposition* (SVD). Suppose $\mathbf{rank}\, A = r \geq k$, and the SVD of A is given by

$$A = U\Sigma V^T,$$

where $U \in \mathbf{R}^{m\times r}$ and $V \in \mathbf{R}^{n\times r}$ are orthogonal matrices, and the matrix $\Sigma = \mathbf{diag}(\sigma_1,\ldots,\sigma_r) \in \mathbf{R}^{r\times r}$ is diagonal with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$ being the *singular values* of A. Then, *a* solution of the PCA problem (1.7) is given by

$$X^\star = U_k\Sigma_k^{1/2} \quad \text{and} \quad Y^\star = \Sigma_k^{1/2}V_k^T, \tag{1.8}$$

where $U_k \in \mathbf{R}^{m\times k}$ and $V_k \in \mathbf{R}^{n\times k}$ are the matrices formed by the first k columns of U and V, respectively, and $\Sigma_k = \mathbf{diag}(\sigma_1, \ldots, \sigma_k) \in \mathbf{R}^{k\times k}$. However, the inverse problem formulation (1.7) is often more convenient when we want to extend PCA to include additional prior information.

Regularized PCA

It is easily seen that the solution of PCA problem given by (1.8) is not unique: For any invertible matrix $W \in \mathbf{R}^{k\times k}$, the pair $(X^\star W, W^{-1}Y^\star)$ is also a solution of (1.7), since $(X^\star W)(W^{-1}Y^\star) = X^\star Y^\star$. To limit the number of PCA solutions, we can add quadratic regularization on X and Y to the PCA objective, resulting in the *quadratically regularized PCA* problem:

$$\text{minimize} \quad \|XY - A\|_F^2 + \lambda(\|X\|_F^2 + \|Y\|_F^2), \tag{1.9}$$

where $\lambda > 0$ is the regularization coefficient. Similar to ridge regression, the quadratic regularization terms encourage the entries of X and Y to be small. Although solutions of quadratically regularized PCA are still not unique, the solution set of (1.9) is significantly smaller than that of the original PCA problem (1.7): If $(X^\star, Y^\star)$ is a solution of (1.9), then so is $(X^\star Q, Q^T Y^\star)$ for any orthogonal matrix $Q \in \mathbf{R}^{k\times k}$.

As another example, similar to lasso regression, we can promote sparsity in the solution of PCA by solving the problem:

$$\text{minimize} \quad \|XY - A\|_F^2 + \lambda(\|X\|_{\text{sav}} + \|Y\|_{\text{sav}}), \tag{1.10}$$

where $\lambda > 0$ is the regularization coefficient, and $\|\cdot\|_{\text{sav}}$ is the *sums of absolute values* norm, defined as

$$\|U\|_{\text{sav}} = \sum_{i=1}^{m}\sum_{j=1}^{n} |U_{ij}|$$

for any matrix $U \in \mathbf{R}^{m\times n}$, which is simply the ℓ_1-norm of U when viewed as a vector in $\mathbf{R}^{mn}$.

Constrained PCA

There are many famous constrained PCA variations that appear frequently in practice, although they might not explicitly named as such. The *sparse dictionary learning* problem corresponds to PCA with sparsity regularization on the matrix Y, and a Frobenius norm constraint on the matrix X:

$$\begin{array}{ll} \text{minimize} & \|XY - A\|_F^2 + \lambda\|Y\|_{\text{sav}} \\ \text{subject to} & \|X\|_F \leq 1, \end{array}$$

where $\lambda > 0$ is the regularization coefficient. As another example, the *nonnegative matrix factorization* problem corresponds to PCA with the constraint that both X and Y are componentwise nonnegative:

$$\begin{array}{lll} \text{minimize} & \|XY - A\|_F^2 & \\ \text{subject to} & X_{ij} \geq 0, \quad i = 1, \ldots, m, & j = 1, \ldots, k \\ & Y_{ij} \geq 0, \quad i = 1, \ldots, k, & j = 1, \ldots, n. \end{array}$$

All these examples suggests that by formulating machine learning models as inverse problems, it is often straightforward to incorporate various types of prior information and structural assumptions on the model, depending on specific application needs.

1.3 Solving inverse problems

From a technical perspective, we may state without loss of generality that fitting a machine learning model corresponds to solving a mathematical optimization problem in the form

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \ i = 1, \ldots, m \\ & h_i(x) = 0, \ i = 1, \ldots, p, \end{array} \tag{1.11}$$

since any regularization function can be absorbed as an (implicit) secondary objective into f_0.

1.3.1 Convex optimization

An inverse problem of the form (1.11) is called a *convex optimization problem* (or *convex program*) if $f_i \colon \mathbf{R}^n \to \mathbf{R}$ are convex functions for all $i = 0, \ldots, m$, *i.e.*, satisfy

$$f_i(\theta x + (1-\theta)y) \leq \theta f_i(x) + (1-\theta) f_i(y)$$

for all $x, y \in \mathbf{R}^n$ and $\theta \in [0, 1]$, and $h_i \colon \mathbf{R}^n \to \mathbf{R}$ are affine functions for all $i = 1, \ldots, p$. The least squares problem (1.4) and its variations presented in §1.2.1 are all convex optimization problems. As a special case, if the objective and constraint functions of (1.11) are all affine, then the problem is called a *linear program*. Linear programs are, of course, convex optimization problems.

There is no general analytical solution for convex optimization problems, but there are very effective and reliable methods for solving them, for example, interior-point methods, which can solve convex optimization problems of moderate size (with up to a few thousands of variables and constraints) to high accuracy within at most a few tens of seconds on a modern desktop computer.

Nowadays, we can make the claim that solving general convex optimization problems is a mature technology, just like solving least squares problems. In other words, if an inverse problem can be recognized, formulated, or transformed as a convex optimization problem, then we can simply treat it as already being solved. (The related techniques are covered in chapter 2.)

1.3.2 Nonlinear optimization

Nonlinear optimization problems (or *nonlinear programs*) are optimization problems where the objective or constraint functions are not affine, but not known to be convex. The PCA problem (1.7) and its variants presented in §1.2.2 are examples of nonlinear optimization problems.

With only few exceptions (such as ordinary PCA), unfortunately, there is no efficient method for solving (*i.e.*, finding a global minimum of) general nonlinear optimization problems, and even simple looking problems with as few as ten variables can be extremely challenging or even intractable. Therefore, nonlinear optimization problems are usually handled under some kind of compromise, such as time or solution quality.

Local optimization

Local optimization methods redefine (or in other words, *relax*) the notion of 'solving' an optimization problem as finding a point that only achieves *local minimum*. In other words, the compromise is to give up seeking a point among all feasible points that achieves the lowest objective value, but instead, we only seek a point that achieves the lowest objective value among all feasible points in a small neighborhood around it.

Obviously, there are several disadvantages of local optimization methods, beyond the obvious one that it is not guaranteed to find a global minimum. The local solution is often sensitive to the choice of the initial point, and different initial points may lead to very different final results. Besides, it is often difficult to assess the quality of a local solution, since there is usually no information about how far the local solution is from the global minimum.

Nevertheless, in terms of handling the (nonconvex) inverse problems of machine learning models, local optimization methods are often sufficient in practice, since in many applications, a locally optimal 'solution' is already good enough for the purpose of model fitting. Moreover, these methods can be fast and scale well to large problems. As a result, local optimization methods are well developed and widely used in the machine learning community. Several local optimization methods that are useful in handling disciplined machine learning problems are presented in chapter 3.

Global optimization

Global optimization methods, as the name suggests, aim to find a global minimum of a nonlinear optimization problem, or at least provide some guarantee on the optimality of the solution found; the compromise is then the computational cost, *i.e.*, time. For some problems with special structure, there exist global optimization methods that can find a solution within reasonable time. The worst-case computational complexity of these methods is, however, exponential in the problem size (*i.e.*, the dimension of variable and the number of constraints). In general, global optimization methods may take an impractically long time to find a solution, even for small problems with a few tens of variables.

In practice, global optimization is used for problems with a small number of variables, where the computing time is not a major concern, and finding a globally optimal point is critical, such as in safety-critical applications. To the best of our knowledge, although global optimization methods do exist, they do not appear so frequently in machine learning, probably due to the typically large size of the inverse problems of machine learning models and the fact that locally optimal points are

often sufficient for most applications in this field.

1.4 Outline

This book is divided into three main parts, titled *Optimization*, *Disciplined modules*, and *Applications*, along with several appendices that include some background and supplementary material.

1.4.1 Part I: Optimization

Part I covers the basics of mathematical optimization, with the aim of getting the reader familiar with those terminologies, methods, and the philosophy of representing machine learning models in the form of inverse problems that have known structures. The contents can be roughly divided into two categories based on the convexity of the optimization problems being dealt with.

Chapter 2 introduces convex sets, convex functions, and convex optimization problems. Its goal is two fold: to provide the background needed for the rest of the book, and to introduce *constructive convex analysis*, a way to build complex convex sets and functions from a small collection of *atomic* ones using *convexity-preserving operations*, so that convexity can be checked automatically. These ideas help readers recognize when the inverse problems of some machine learning models can be formulated as convex programs, so that it is immediately clear that these problems can be solved robustly and efficiently.

Chapter 3 introduces *sequential convex programming* methods for *approximately* solving nonlinear optimization problems. As the name suggests, these are heuristic approaches that solve a sequence of convex approximations of the original nonconvex problem. The aim is to provide a practical toolbox for handling (especially constrained) nonconvex problems that arise in machine learning, at least as a first attempt. Appendix C complements the chapter with technical details that are useful for implementing these ideas robustly. Our treatment is necessarily simplified compared to the vast literature on nonlinear optimization, but it is intended to be sufficient for readers to develop working implementations. We are sure that the material here will make experts in nonlinear optimization feel oversimplified, and we apologize to them in advance.

1.4.2 Part II: Disciplined modules

Part II divides the inverse problem of different machine learning models into several disciplined modules, each corresponding to a specific aspect of the modeling procedure, that says, *Objectives*, *Regularization functions*, and *Constraints*. The modules covered in this part serve as fundamental building blocks for constructing more complicated machine learning models, according to specific application needs.

Chapter 4 discusses the fundamental objectives corresponding to different types of machine learning tasks. These objectives can be roughly categorized into three types: *approximation*, *statistical estimation*, and *discrimination*.

Chapter 5 introduces regularization techniques. We start from some basic theories about multiobjective optimization in §5.1, and then present a variety of regularization functions that are commonly used in practice, along with their properties and the intuition behind them.

Material presented in chapter 6 consists of two topics: The first half of the chapter is about some generic examples of constraints that appear in different types of machine learning models. The second half discusses *relaxation* techniques for operating on constrained optimization problems, which are often used to make the original problem easier to solve, at the cost of some loss of solution quality. Here we will also cover the well-known method of *Lagrangian relaxation* and *duality* of constrained optimization.

1.4.3 Part III: Applications

Part III introduces several examples demonstrating how the materials presented in the previous parts can be applied to construct more complicated machine learning models according to different prior knowledge and application needs. Several examples presented here are widely known and used in practice, but their inverse problem formulations are not obvious or less known. Other examples may have simple underlying ideas and clean inverse problem formulations, but analyzing and transforming these problems into tractable forms requires nontrivial effort.

Chapter 7 discusses an important extension of standard machine learning inverse problems where uncertainty is present in the data, and the goal is to find a solution that is robust against this uncertainty.

Chapter 8 is about fitting machine learning models that involve latent variables or factors, which are not directly observed but are to be inferred from the data. Based on the continuity of the latent factors, we can roughly categorize these models into two types: mixture models (with the problem of clustering as a special case) and factorization models (*i.e.*, low rank matrix approximation and principal component analysis). We will also cover some generalization of these ideas.

1.4.4 Appendices

There are three appendices attached to the end of this book. Appendix A provides some mathematical background material that is useful to review but not strictly necessary for understanding the main contents of this book. It can also be considered as some expanded text for setting up our notation.

Appendix B more or less deals with some real practical aspects of convex optimization, *i.e.*, how to specify optimization problems in a way that can be understood and processed by a computer for automatic convexity verification. This appendix presents the framework of *disciplined convex programming*, which is based on constructive convex analysis and the related material introduced in chapter 2. It is the foundation of many *domain specific computer languages* for specifying, canonicalizing, and solving convex optimization problems. There are many excellent computer softwares that implement the disciplined convex programming framework, but we only present the general idea and the basic rules here, without going into details of any specific software implementation.

Appendix C discusses some technical issues related to implementing a sequential convex programming solver, as some complementary materials to chapter 3. The topics addressed here are largely generic and arise independently of specific applications. Nevertheless, many additional issues, which may vary wildly across different problems, might still need to be handled to ensure that the algorithm works properly and robustly, and we will never try to make this appendix encyclopedic on these topics.

1.4.5 Exercises

At the end of most chapters, there are several exercises for the reader to practice the materials covered in the chapter. Some exercises are straightforward applications of the concepts and methods presented in the chapter, while others may require some additional thinking and research. In other words, the difficulty level of the exercises is mixed, and varies without warning from quite straightforward to rather tricky.

Many exercises involve showing or establishing some statements or claims, which might appear in the main text of the chapter without proof, or may be completely new. These material can be considered as some supplementary contents to the chapter, and could possibly appear to be quite useful in various real world applications. Some exercises may require the use of a computer and programming in an appropriate high-level language to carry out numerical experiments.

1.5 Notation

In this section, we describe the basic notation used throughout this book, which is more or less standard, or have become standard after years of evolution in different communities; a more comprehensive list appears on page 381.

Some specific sets

We use $\mathbf{R}$ to denote the set of real numbers, $\mathbf{R}_+$ for the set of nonnegative real numbers, and $\mathbf{R}_{++}$ for the set of positive real numbers. Similarly, $\mathbf{Z}$ denotes the set of integers, and $\mathbf{Z}_+$ and $\mathbf{Z}_{++}$ denote the set of nonnegative and positive integers, respectively. All these symbols may be extended to denote the set of vectors or matrices with entries from these sets: For example, $\mathbf{R}^n$ denotes the set of real n-vectors, and $\mathbf{R}^{m\times n}$ denotes the set of real $m \times n$ matrices. We use $\mathbf{S}^n$ to denote the set of symmetric $n \times n$ matrices, $\mathbf{S}^n_+$ for the set of symmetric positive semidefinite $n \times n$ matrices, and $\mathbf{S}^n_{++}$ for the set of symmetric positive definite $n \times n$ matrices.

Vectors and matrices

Vectors and matrices are delimited with square brackets, with the entries separated by space. We use parentheses to construct column vectors from comma separated lists. For example, suppose $a, b, c \in \mathbf{R}$, the following representations for a vector in

$\mathbf{R}^3$ are equivalent:

$$(a, b, c) = \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} a & b & c \end{bmatrix}^T.$$

The symbol 0 denotes a vector or matrix with all entries equal to zero, $\mathbf{1}$ denotes a vector with all entries equal to one, and I denotes the identity matrix. The dimension of these objects should be determined from the context or the text. The notation x_i can refer to the ith entry of the vector x, or to the ith element of a set or sequence of vectors $x_1, x_2, \ldots$, and, again, the context, or the text, makes it clear which is meant.

Generalized inequalities

We use $\leq$ and $<$ to denote the usual scalar inequalities and strict inequalities between real numbers. The squiggly symbols $\preceq$ and $\prec$ denote generalized (strict) inequalities between vectors or matrices. If $x, y \in \mathbf{R}^n$, then $x \preceq y$ means that the inequality holds *componentwise*, *i.e.*,

$$x_i \leq y_i, \quad i = 1, \ldots, n,$$

or in other words, $y - x \in \mathbf{R}^n_+$; similarly, $x \prec y$ means that $y - x \in \mathbf{R}^n_{++}$. If $X, Y \in \mathbf{S}^n$, then $X \preceq Y$ means that the matrix $Y - X$ is positive semidefinite, *i.e.*, $Y - X \in \mathbf{S}^n_+$, and $X \prec Y$ means that the matrix $Y - X$ is positive definite, *i.e.*, $Y - X \in \mathbf{S}^n_{++}$. These (partial) orderings for matrices are called the *matrix inequality*.

Functions

Our notation for functions is fairly standard, with one exception: By writing $f\colon \mathbf{R}^n \to \mathbf{R}^m$, we really (or implicitly) mean $f\colon \mathbf{R}^n \to \mathbf{R}^m \cup \{\pm\infty\}$, *i.e.*, $f(x)$ is some vector in $\mathbf{R}^m$ for x in its (effective) *domain* $\mathbf{dom}\, f \subseteq \mathbf{R}^n$, which can be a *subset* of $\mathbf{R}^n$, and $f(x)$ takes either ∞ or $-\infty$ for all $x \notin \mathbf{dom}\, f$. For convex and concave functions (see §2.3.1), this means they are implicitly extended-valued, according to the convention: For all $x \notin \mathbf{dom}\, f$,

$$f(x) = \begin{cases} \infty, & f \text{ is convex} \\ -\infty, & f \text{ is concave.} \end{cases}$$

As an example, the *indicator function* of a set $C \subseteq \mathbf{R}^n$ is written as $I_C\colon \mathbf{R}^n \to \mathbf{R}$, defined as

$$I_C(x) = \begin{cases} 0, & x \in C \\ \infty, & \text{otherwise} \end{cases}$$

with $\mathbf{dom}\, I_C = C$. As another example, the logarithm function is written as $\log\colon \mathbf{R} \to \mathbf{R}$, with $\mathbf{dom} \log = \mathbf{R}_{++}$, meaning that $\log(x)$ is a real number for $x > 0$,

and $\log(x) = -\infty$ for $x \leq 0$. As a more complex example, the *log-determinant* function $f\colon \mathbf{S}^n \to \mathbf{R}$ is defined as

$$f(X) = \begin{cases} \log\det X, & X \in \mathbf{S}^n_{++} \\ -\infty, & \text{otherwise} \end{cases}$$

with $\mathbf{dom}\, f = \mathbf{S}^n_{++}$.

Vector spaces

In most cases, the results and statements in this book are provided under a generic finite-dimensional vector space $\mathbf{R}^n$, with the standard inner product denoted as

$$x^T y = x_1 y_1 + x_2 y_2 + \cdots + x_n y_n,$$

for $x, y \in \mathbf{R}^n$. We will also encounter other vector spaces, such as the space of $k \times k$ symmetric matrices $\mathbf{S}^k$. By specifying a proper basis for a vector space, we can always identify it with $\mathbf{R}^n$. For example, the space of $k \times k$ symmetric matrices $\mathbf{S}^k$ can be identified as $\mathbf{R}^{k(k+1)/2}$, since a $k \times k$ symmetric matrix is determined by its $k(k+1)/2$ distinct entries. We usually leave it to the reader to translate general results or statements on $\mathbf{R}^n$ to other vector spaces.

As a specific example, consider the following statement:

> *Any linear function $f\colon \mathbf{R}^n \to \mathbf{R}$ can be represented as $f(x) = a^T x$, where $a \in \mathbf{R}^n$.*

By choosing a proper basis, we can translate it to the space of $k \times k$ symmetric matrices $\mathbf{S}^k$ as:

> *Any linear function $f\colon \mathbf{S}^k \to \mathbf{R}$ can be represented as $f(X) = \mathbf{tr}(AX)$, where $A \in \mathbf{S}^k$.*

(See also §A.3.1.) As another example, the following statement:

> *The set $\{x \in \mathbf{R}^n \mid a^T x \geq b\}$, where $a \in \mathbf{R}^n$ and $a \neq 0$, defines a halfspace in $\mathbf{R}^n$.*

can be translated to $\mathbf{S}^k$ as:

> *The set $\{X \in \mathbf{S}^k \mid z^T X z \geq 0\}$, where $z \in \mathbf{R}^k$ and $z \neq 0$, defines a halfspace in $\mathbf{S}^k$;*

see example 2.6.

Bibliographical notes

Machine learning

The term *machine learning* was first introduced by Arthur Samuel in 1959 in the field of computer gaming and artificial intelligence [Sam59], and is still a very active research area. Classic textbooks for the related topics include [Bis06], [Mur12], [Zho21], [Mur22], and [Mur23], just list a few.

Deep learning, as a subfield of machine learning, has gained significant popularity in recent years due to its success in various applications (actually, almost everywhere). At the moment, roughly speaking, deep learning is more art than technology, which relies heavily on heuristics and trial-and-error. We will not cover it in this book, but refer interested readers to [GBC16].

Least squares

Least squares is a very old topic, with origins dating back to Gauss in the 1820s in a treatise originally written in Latin and later translated by Steward [Gau95]. Numerical methods for solving least squares problem are covered in the books by Lawson and Hanson [LH95] and Björck [Bjö96], and many other more recent textbooks on linear algebra; see the references in appendix A.

The idea of Tikhonov regularization dates back to the works by Tikhonov starting from the 1940s (originally written in Russian), and Phillips [Phi62], while the name itself was widely known from the book by Tikhonov and Arsenin [TA77] published in 1977, on the theory and applications of regularization for solving ill-posed problems. After Hoerl and Kennard [HK70], Tikhonov regularization is known in the statistical community as ridge regression.

Least squares with the ℓ_1-regularizer was first developed in geophysics in 1986 [SS86], and rediscovered in statistics and popularized as lasso regression by Tibshirani in 1996 [Tib96].

Materials on general regularized and constrained least squares can be found, *e.g.*, in [HTF09, chapter 3] and [BV18, part III].

Principal component analysis

Principal component analysis was first invented in statistics by Pearson in 1901 [Pea01], and was later independently developed and named by Hotelling in the 1930s [Hot33, Hot36]. PCA has been assigned to different names in different domains, such as *Karhunen-Loève transform* in signal processing, *Hotelling transform* in multivariate quality control and *proper orthogonal decomposition* in mechanical engineering. In numerical linear algebra, PCA of some matrix $A \in \mathbf{R}^{m\times n}$ is closely related to *singular value decomposition* of A and *eigenvalue decomposition* (or *spectral decomposition*) of A^TA; see, *e.g.*, [Str06, chapters 5 and 6] and [Mey23, chapters 5 and 7]. The book by Jolliffe [Jol02] is a classic reference on PCA and its applications.

PCA also has a long history in matrix analysis and approximation theory, evolving in parallel with developments in the statistics community. The inverse problem formulation of PCA, given by (1.6) and (1.7), was originally proposed as low rank approximation problems (in a much general form) and solved by Schmidt [Sch07], and later rediscovered by Eckart and Young [EY36]. Mirsky [Mir60] provided a more general result on the best approximation of matrices under unitarily invariant norms. Therefore, the PCA problem

solution via SVD is also referred to as the *matrix approximation lemma* or *Eckart-Young-Mirsky theorem*.

Regularized and constrained PCA variations have been extensively studied in the literature, as *generalized low rank models*, especially in recent years due to their wide applications in machine learning and data analysis. Interested readers may refer to the monograph by Udell *et al.* [UHZB16] for a comprehensive review, as well as the references listed in chapter 8.

Convex optimization

Convex optimization is a mature research area with a solid theoretical foundation and effective numerical methods. The book by Boyd and Vandenberghe [BV04] is a classic reference in this field. Some other references can be found in chapter 2.

Convex analysis, the mathematics of convex sets, functions, and optimization problems, is a well developed subfield of mathematics, and serves as the theoretical foundation of convex optimization. Basic references of these topics can be found in [Roc70], [HL93a], [HL93b], [HL01], [BNO03], and [BL06].

Solution methods of convex optimization problems, *i.e.*, algorithms for solving convex optimization problems, include *interior-point methods* [NN94, PW00, NT08], *subgradient methods* [Sho85, Sho98, BXM03, Boy14], *bundle methods* [HL93b], *cutting-plane methods* [Kel60, EM75, GLY96], and the *ellipsoid method* [BGT81, Sho98]. For large-scale convex optimization problems, *operator splitting methods* [BC17, RY22], such as *proximal methods* [PB14] and the *alternating direction method of multipliers* [BPC+11], often works quite efficiently in practice.

Many softwares, or specifically, *domain specific languages*, have been developed for specifying convex optimization problems in a natural, human-readable way, which allow automatic verification of the convexity of an optimization problem specified by the user, and then transform it into standard forms that can be directly handled by numerical solvers. Such modeling framework of convex optimization problems exists for many programming languages, include YALMIP [Lof04] and CVX [GB14] for MATLAB, Convex.jl [UMZ+14] for Julia, CVXPY [DB16, AVDB18] for Python, and CVXR [FNB20] for R.

Nonlinear optimization

There are many good materials on local optimization methods for nonlinear optimization, including Nocedal and Wright [NW06], Luenberger and Ye [LY08], Bertsekas [Ber16], and Gill *et al.* [GMW19]. References for some speficic methods, such as *alternate convex search*, *convex-concave procedure*, and *sequential convex approximation*, can be found in chapter 3.

Many textbooks provide useful material on global optimization methods, such as [TŽ89], [HP95], [HT96], [HPT00], [PR02], [Wei09], and [LS13]. In particular, convex optimization has been used for finding bounds of nonconvex problems, some examples can be found in the books above on global optimization, the book by Ben-Tal and Nemirovski [BN01, §4.3], and the review paper by Nesterov *et al.* [NWY00].

Similar to convex optimization, several domain specific languages have been developed for specifying and solving nonlinear optimization problems, such as AMPL [FGK90], GAMS [BKMR98], JuMP [DHL17], CasADi [AGH+19], and Pyomo [BHH+21].

Notation

Generalized inequalities can be formally defined via (proper) *convex cones* (see §2.1.2, page 25). In particular, the componentwise inequality on $\mathbf{R}^n$ are defined with respect to the nonnegative orthant $\mathbf{R}^n_+$ (example 2.1), and the matrix inequality on $\mathbf{S}^n$ are defined with respect to the positive semidefinite cone $\mathbf{S}^n_+$ (§2.1.3, page 30). We refer interested readers to [BV04, §2.4, §2.6, and §3.6], as well as to the references listed at the end of those chapters, for formal definitions and more discussions about generalized inequalities and their properties.

Our notation is mostly adapted from the books [BV04] and [BV18] by Boyd and Vandenberghe, which has (if not yet by the time when they were published) become more or less standard in optimization and related fields. Some related concepts, such as effective domain and extended-value extension of functions, which are directly used here without definition, can be found in [Roc70, §4] and [BV04, §3.1.2].

Part I

Optimization

Chapter 2

Convex optimization

2.1 Convex sets

2.1.1 Affine sets

Lines and line segments

Let $x_1, x_2 \in \mathbf{R}^n$ and $x_1 \neq x_2$. Points of the form

$$y = \theta x_1 + (1 - \theta)x_2 \tag{2.1}$$

with $\theta \in \mathbf{R}$ form a *line* passing through x_1 and x_2. In particular, if the parameter $\theta = 1$, we have $y = x_1$, and $\theta = 0$ corresponds to $y = x_2$. If $\theta \in [0, 1]$, then the set of points y, given by

$$\{\theta x_1 + (1 - \theta)x_2 \mid 0 \leq \theta \leq 1\},$$

form the (closed) *line segment* between x_1 and x_2.

We can interpret lines and line segments by expressing (2.1) as

$$y = x_2 + \theta(x_1 - x_2),$$

which shows that y is obtained by starting at the *base point* x_2 and moving in the *direction* of the vector $x_1 - x_2$ (which points from x_2 to x_1) scaled by the number θ. Thus, the parameter θ can be interpreted as the *fraction* of the way from x_2 to x_1 where y lies. If $\theta < 0$, then y lies on the line that extends beyond x_2 in the direction away from x_1. As θ increases from 0 to 1, the point y moves from x_2 to x_1. If $\theta > 1$, then y lies on the line that extends beyond x_1 in the direction away from x_2. This interpretation of lines and line segments is illustrated in figure 2.1.

Affine sets

A set $C \subseteq \mathbf{R}^n$ is *affine* if the line through any two distinct points in C lies in C, *i.e.*, for all $x_1, x_2 \in C$ and $\theta \in \mathbf{R}$, we have

$$\theta x_1 + (1 - \theta)x_2 \in C.$$

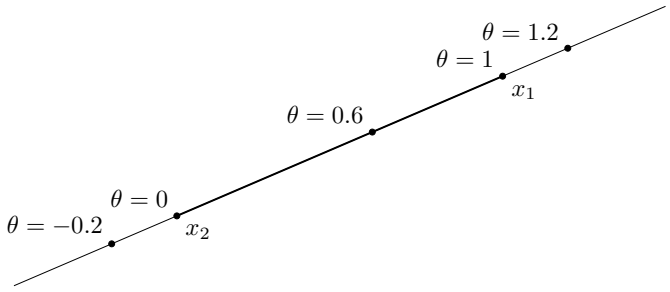

Figure 2.1 The line passing through x_1 and x_2 described by (2.1) with parameter $\theta \in \mathbf{R}$. The line segment between x_1 and x_2, which corresponds to $\theta \in [0, 1]$, is shown thicker.

In other words, an affine set contains the linear combination of any two of its points, provided the coefficients of the linear combination sum to one.

This definition (or property) of affine sets can be extended to more than two points. An *affine combination* of points $x_1, \ldots, x_k \in \mathbf{R}^n$ is a linear combination of the form

$$\theta_1 x_1 + \cdots + \theta_k x_k$$

where the coefficients $\theta \in \mathbf{R}^k$ satisfy

$$\mathbf{1}^T \theta = \theta_1 + \cdots + \theta_k = 1.$$

According to the previous definition of affine sets, it can be easily shown by induction that an affine set contains every affine combination its points, *i.e.*, if C is an affine set, then for all $x_1, \ldots, x_k \in C$ and $\theta \in \mathbf{R}^k$ with $\mathbf{1}^T \theta = 1$, we have

$$\theta_1 x_1 + \cdots + \theta_k x_k \in C.$$

2.1.2 Convex sets and cones

Convex sets

A set C is *convex* if the line segment between any two points in C lies in C, *i.e.*, for all $x_1, x_2 \in C$ and $\theta \in [0, 1]$, we have

$$\theta x_1 + (1 - \theta) x_2 \in C.$$

Every affine set is also convex, since the definition of affine sets requires that the entire line through any two points in the set lies in the set, which therefore includes the line segment between the two points. Some simple examples of convex and nonconvex sets in $\mathbf{R}^2$ are shown in figure 2.2.

Convex combinations

A *convex combination* of points $x_1, \ldots, x_k \in \mathbf{R}^n$ is a linear combination of the form

$$\theta_1 x_1 + \cdots + \theta_k x_k$$

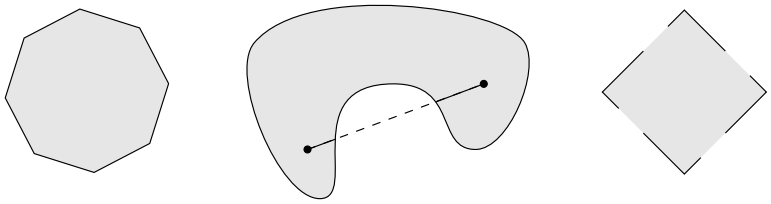

Figure 2.2 Examples of convex and nonconvex sets in $\mathbf{R}^2$. *Left.* The octagon includes its boundary (shown darker) is a convex set. *Middle.* The kidney shaped set is not convex, since a part of the line segment between the two points in the set shown as dots is not contained in the set. *Right.* The diamond with some but not all of its boundary removed is not convex.

where the coefficients $\theta \in \mathbf{R}^k$ satisfy

$$\mathbf{1}^T\theta = 1 \quad \text{and} \quad \theta \succeq 0.$$

(Recall that $\theta \succeq 0$ means $\theta_i \geq 0$ for all $i = 1, \ldots, k$). As with affine sets, it can be shown that a convex set contains every convex combination of its points, *i.e.*, if C is a convex set, then for all $x_1, \ldots, x_k \in C$ and $\theta \in \mathbf{R}^k$ with $\mathbf{1}^T\theta = 1$ and $\theta \succeq 0$, we have

$$\theta_1 x_1 + \cdots + \theta_k x_k \in C.$$

Convex combinations are sometimes interpreted as a *weighted average* or *mixture* of the points $x_1, \ldots, x_k$, where the fraction of the point x_i is given by the coefficient θ_i.

The *convex hull* of a set C, which we denote as $\mathbf{conv}\, C$, is the set of all convex combinations of the points in C, *i.e.*,

$$\mathbf{conv}\, C = \left\{ \sum_{i=1}^{k} \theta_i x_i \;\middle|\; \begin{array}{ll} k \in \mathbf{Z}_{++} & \\ x_i \in C, & i = 1, \ldots, k \\ \mathbf{1}^T\theta = 1, & \theta \succeq 0 \end{array} \right\}.$$

It can be easily shown that the convex hull of any set C is convex, and $\mathbf{conv}\, C$ is the smallest convex set that contains C, *i.e.*, if B is a convex set with $C \subseteq B$, then $\mathbf{conv}\, C \subseteq B$. This is illustrated in figure 2.3.

Convex combinations of infinite points

The properties of convex combinations of finite points in a convex set can be generalized to infinite points, including infinite sums, integrals, and in the most general form, probability distributions. To describe this, let $C \subseteq \mathbf{R}^n$ be a convex set.

Suppose $x_1, x_2, \ldots \in C$ and $\theta_1, \theta_2, \ldots \in \mathbf{R}$ satisfy

$$\sum_{i=1}^{\infty} \theta_i = 1 \quad \text{and} \quad \theta_i \geq 0,\ i = 1, 2, \ldots,$$

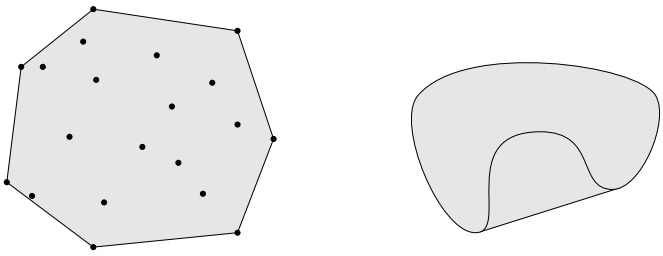

Figure 2.3 The convex hull of two sets in $\mathbf{R}^2$. *Left.* The convex hull (heptagon including its boundary, shown shaded) of a set of twenty points (shown as dots). *Right.* The convex hull of the kidney shaped set in figure 2.2 is shown as the shaded set.

then we have

$$\sum_{i=1}^{\infty} \theta_i x_i \in C$$

if the series converges.

More generally, let $p\colon \mathbf{R}^n \to \mathbf{R}$ be a function satisfying

$$\int_C p(x)\ dx = 1 \quad \text{and} \quad p(x) \geq 0 \text{ for all } x \in C,$$

then we have

$$\int_C p(x)x\ dx \in C$$

if the integral exists.

In the most general form, let $x \in \mathbf{R}^n$ be a random vector with

$$\mathbf{prob}(x \in C) = 1,$$

then we have

$$\mathbf{E}\, x \in C$$

if the expectation exists. This includes, for example, the case of convex combinations of finite points in the following sense: Let $x \in \{x_1, \ldots, x_k\}$ be a discrete random vector with

$$\mathbf{prob}(x = x_i) = \theta_i, \quad i = 1, \ldots, k.$$

Since the probabilities of all possible outcomes of x must sum to one, the vector $\theta \in \mathbf{R}^k$ satisfies $\mathbf{1}^T\theta = 1$ and $\theta \succeq 0$. Then, we have

$$\mathbf{E}\, x = \theta_1 x_1 + \cdots + \theta_k x_k,$$

which is a convex combination of the k points $x_1, \ldots, x_k$ as defined above.

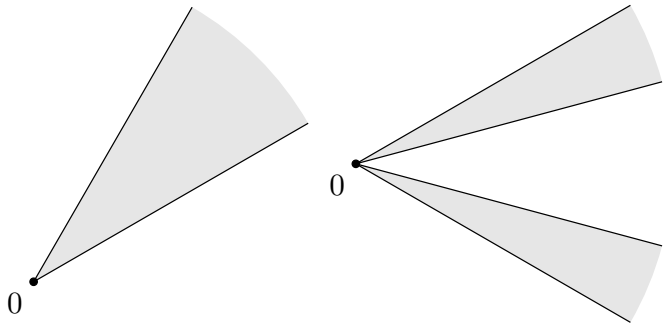

Figure 2.4 Examples of convex (*left*) and nonconvex (*right*) cones in $\mathbf{R}^2$.

Cones

A set C is called a *cone* if for all $x \in C$ and $\theta \geq 0$, we have

$$\theta x \in C.$$

By this definition, a cone should include the origin, since for any $x \in C$, we have $0 \cdot x = 0 \in C$. A set C is a *convex cone* if for all $x_1, x_2 \in C$ and $\theta_1, \theta_2 \geq 0$, we have

$$\theta_1 x_1 + \theta_2 x_2 \in C.$$

Some examples of convex and nonconvex cones in $\mathbf{R}^2$ are shown in figure 2.4.

A *conic combination* of points $x_1, \ldots, x_k \in \mathbf{R}^n$ is a linear combination of the form

$$\theta_1 x_1 + \cdots + \theta_k x_k$$

where the coefficients $\theta \in \mathbf{R}^k$ satisfy

$$\theta \succeq 0.$$

If C is a convex cone, then for all $x_1, \ldots, x_k \in C$ and $\theta \in \mathbf{R}^k$ with $\theta \succeq 0$, we have

$$\theta_1 x_1 + \cdots + \theta_k x_k \in C.$$

Similar to convex combinations, this property of conic combinations can be generalized to infinite points.

2.1.3 Examples

We start with some simple examples of convex sets:

- The empty set $\emptyset$, a singleton $\{x_0\}$, and the entire Euclidean space $\mathbf{R}^n$ are all affine subsets of $\mathbf{R}^n$, which are hence convex.
- A line segment between two different points in $\mathbf{R}^n$ is a convex set, but is not affine.

- A line in $\mathbf{R}^n$ is an affine set and hence convex. In particular, if a line passes through the origin, then it is a convex cone.
- A *ray* in $\mathbf{R}^n$, which has the form $\{x_0 + \theta v \mid \theta \geq 0\}$ with $v \neq 0$, is convex. If the ray starts from the origin, *i.e.*, $x_0 = 0$, then it is a convex cone.
- Let $C \subseteq \mathbf{R}^n$ be a convex cone. A *translated cone* of C, given by $x_0 + C$ with $x_0 \in \mathbf{R}^n$, is a convex set, but is not a cone unless $x_0 = 0$.
- A subspace of $\mathbf{R}^n$ is an affine set and a convex cone, and hence also convex.

Solution set of linear equations

Let $A \in \mathbf{R}^{m \times n}$ and $b \in \mathbf{R}^m$. The solution set of a system of linear equations, given by

$$C = \{x \in \mathbf{R}^n \mid Ax = b\},$$

is an affine set in $\mathbf{R}^n$. To show this, we use the definition of affine sets. Let $x_1, x_2 \in C$ be two different points, *i.e.*, $Ax_1 = b$ and $Ax_2 = b$. Then, for any $\theta \in \mathbf{R}$, we have

$$\begin{aligned} A(\theta x_1 + (1-\theta)x_2) &= \theta A x_1 + (1-\theta)Ax_2 \\ &= \theta b + (1-\theta)b \\ &= b, \end{aligned}$$

which shows that the affine combination $\theta x_1 + (1-\theta)x_2$ also lies in C.

The converse is also true (although we will not show this): Every affine set can be expressed as the solution set of a system of linear equations.

Hyperplanes and halfspaces

Consider the solution set of a linear equation of the form

$$C = \{x \in \mathbf{R}^n \mid a^T x = b\},$$

where $a \in \mathbf{R}^n$, $a \neq 0$, and $b \in \mathbf{R}$. The set C is called a *hyperplane* in $\mathbf{R}^n$. According to the previous example, hyperplanes are affine sets and hence convex. To interpret hyperplanes geometrically, we can express C as

$$\{x \mid a^T(x - x_0) = 0\} = x_0 + a^\perp, \tag{2.2}$$

where x_0 is any point in C, *i.e.*, any point that satisfies $a^T x_0 = b$, and

$$a^\perp = \{v \in \mathbf{R}^n \mid a^T v = 0\}$$

denotes the orthogonal complement of a, *i.e.*, the set of all vectors that are orthogonal to a. The expression (2.2) shows that a hyperplane consists of an offset x_0 plus all vectors orthogonal to a. The vector a is called a *normal vector* of the hyperplane, and the constant b determines the offset of the hyperplane from the origin. This geometric interpretation of hyperplanes is illustrated in figure 2.5.

Roughly speaking, a hyperplane divides the Euclidean space $\mathbf{R}^n$ into two *halfspaces*, by replacing the equality in the definition of hyperplanes to an inequality. A

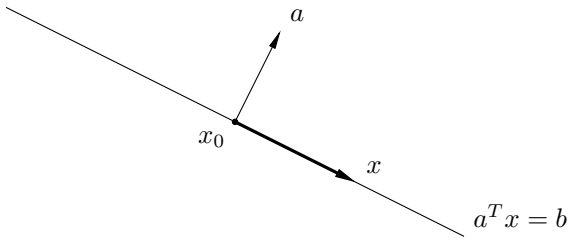

Figure 2.5 A hyperplane in $\mathbf{R}^2$ corresponding to the linear equation $a^T x = b$. The point x_0 lies in the hyperplane, and the vector a is a normal vector of the hyperplane. For any point x in the hyperplane, the vector $x - x_0$ (shown as the thicker arrow) is orthogonal to a.

(closed) halfspace associated with a hyperplane $C = \{x \in \mathbf{R}^n \mid a^T x = b\}$ $(a \neq 0)$ is given by

$$D = \{x \in \mathbf{R}^n \mid a^T x \leq b\},$$

which is the solution set of one linear inequality. Halfspaces are convex sets, but not affine. To show this, we use the definition of convex sets. Let $x_1, x_2 \in D$ be two different points in the halfspace, *i.e.*, $a^T x_1 \leq b$ and $a^T x_2 \leq b$. Then, for any $\theta \in [0, 1]$, we have

$$\begin{aligned} a^T(\theta x_1 + (1-\theta)x_2) &= \theta a^T x_1 + (1-\theta)a^T x_2 \\ &\leq \theta b + (1-\theta)b \\ &= b, \end{aligned}$$

which shows that the line segment between x_1 and x_2 lies in D. Geometrically, we can interprete halfspaces by expressing D in a similar form as in (2.2):

$$\{x \mid a^T(x - x_0) \leq 0\} = x_0 + \{v \mid a^T v \leq 0\},$$

where x_0 is any point in the hyperplane C. This expression shows that a halfspace consists of an offset x_0 plus all vectors that form an nonacute angle with the normal vector a. This interpretation is illustrated in figure 2.6.

Polyhedra

A *polyhedron* is the intersection of a finite number of halfspaces and hyperplanes, which can be expressed as a set of the form

$$P = \left\{ x \in \mathbf{R}^n \;\middle|\; \begin{array}{ll} a_i^T x \leq b_i, & i = 1, \ldots, m \\ c_i^T x = d_i, & i = 1, \ldots, p \end{array} \right\}. \tag{2.3}$$

We can express (2.3) more compactly as

$$P = \{x \in \mathbf{R}^n \mid Ax \preceq b, \quad Cx = d\}, \tag{2.4}$$

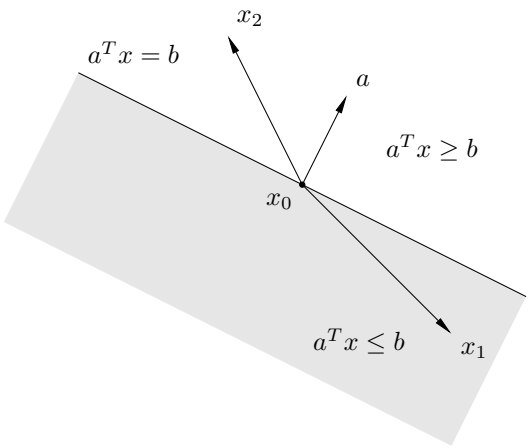

Figure 2.6 Two halfspaces associated with the hyperplane $\{x \mid a^T x = b\}$ in $\mathbf{R}^2$. The halfspace determined by $a^T x \leq b$ (shown shaded) includes all points x_1 where the vector $x_1 - x_0$ makes a nonacute angle with the normal vector a. The other halfspace determined by $a^T x \geq b$ includes all points x_2 where the vector $x_2 - x_0$ makes a nonobtuse angle with the normal vector a.

where

$$A = \begin{bmatrix} a_1^T \\ \vdots \\ a_m^T \end{bmatrix} \in \mathbf{R}^{m \times n} \quad \text{and} \quad C = \begin{bmatrix} c_1^T \\ \vdots \\ c_p^T \end{bmatrix} \in \mathbf{R}^{p \times n}.$$

Using similar arguments as for affine sets and halfspaces, we can easily show that polyhedra are convex sets. Figure 2.7 shows an example of a polyhedron defined as the intersection of five halfspaces in $\mathbf{R}^2$.

Example 2.1 The *nonnegative orthant* is the set of points with nonnegative entries, which is written as

$$\mathbf{R}^n_+ = \{x \in \mathbf{R}^n \mid x \succeq 0\}.$$

The nonnegative orthant is a polyhedron.

Similarly, we denote the set of points with positive entries as

$$\mathbf{R}^n_{++} = \{x \in \mathbf{R}^n \mid x \succ 0\},$$

which is the interior of the nonnegative orthant $\mathbf{R}^n_+$.

Simplexes

Simplexes are a special class of polyhedra. Let $x_0, x_1, \ldots, x_k \in \mathbf{R}^n$ be $k+1$ *affinely independent* points, which means the vectors $x_1 - x_0, \ldots, x_k - x_0$ are linearly independent. The *k-dimensional* simplex C defined by these points is their convex hull, given by

$$C = \mathbf{conv}\{x_0, x_1, \ldots, x_k\} = \{\theta_0 x_0 + \theta_1 x_1 + \cdots + \theta_k x_k \mid \theta \succeq 0,\ \mathbf{1}^T \theta = 1\}.$$

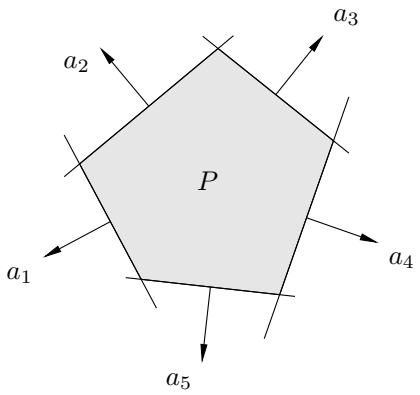

Figure 2.7 The polyhedron $P \subseteq \mathbf{R}^2$ is the intersection of five halfspaces with normal vector $a_1, \ldots, a_5$.

As the definition suggests, simplexes are convex sets. Some basic examples of simplexes includes: A 0-dimensional simplex is a singleton, a 1-dimensional simplex is a line segment between two points, a 2-dimensional simplex is a triangle including its interior defined by three points, and a 3-dimensional simplex is a tetrahedron including its interior defined by four points.

Example 2.2 The *unit simplex* in $\mathbf{R}^n$ is the n-dimensional simplex defined by the zero vector and the standard basis vectors in $\mathbf{R}^n$, *i.e.*, the convex hull of the set $\{0, e_1, \ldots, e_n\} \subseteq \mathbf{R}^n$, which can be expressed as

$$\{x \in \mathbf{R}^n \mid x \succeq 0,\ \mathbf{1}^T x \leq 1\}.$$

Figure 2.8 (left) shows the unit simplex in $\mathbf{R}^3$.

Example 2.3 The *probability simplex* in $\mathbf{R}^n$ is the $(n-1)$-dimensional simplex defined by the standard basis vectors in $\mathbf{R}^n$, which can be expressed as

$$\mathbf{conv}\{e_1, \ldots, e_n\} = \{x \in \mathbf{R}^n \mid x \succeq 0,\ \mathbf{1}^T x = 1\}. \tag{2.5}$$

Geometrically, the probability simplex (2.5) corresponds to the face of the unit simplex in $\mathbf{R}^n$ that is not parallel to the coordinate planes. Vectors in the probability simplex (2.5) correspond to probability distributions on a set with n elements, with x_i interpreted as the probability of the ith element.

Figure 2.8 (right) shows the probability simplex in $\mathbf{R}^3$.

Norm balls

Let $\|\cdot\|$ be a norm on $\mathbf{R}^n$ (see §A.3.2). A *norm ball* associated with the norm $\|\cdot\|$ is the set

$$B = \{x \in \mathbf{R}^n \mid \|x - x_0\| \leq r\},$$

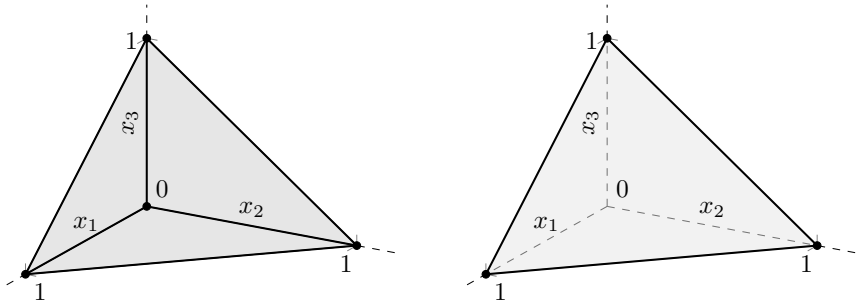

Figure 2.8 The unit simplex (*left*) and the probability simplex (*right*) in $\mathbf{R}^3$. The vertices of both simplexes are shown as dots, and the edges are shown thicker.

where $r > 0$ is the *radius* and $x_0 \in \mathbf{R}^n$ is the *center*. Norm balls are convex sets: If $x_1, x_2 \in B$, *i.e.*, $\|x_1 - x_0\| \leq r$ and $\|x_2 - x_0\| \leq r$, then, for any $\theta \in [0, 1]$, we have

$$\begin{aligned} \|\theta x_1 + (1-\theta)x_2 - x_0\| &= \|\theta(x_1 - x_0) + (1-\theta)(x_2 - x_0)\| \\ &\leq \theta\|x_1 - x_0\| + (1-\theta)\|x_2 - x_0\| \\ &\leq \theta r + (1-\theta)r \\ &= r, \end{aligned}$$

where the first inequality is from the triangle inequality and the positive homogeneity of norms. Figure 2.9 shows some examples of norm balls centered at the origin with radius one in $\mathbf{R}^2$ for different norms (which are sometimes called the *unit balls*).

Example 2.4 A *Euclidean ball* (or just *ball*) in $\mathbf{R}^n$ has the form

$$\begin{aligned} \mathcal{B}(x_0, r) &= \{x \in \mathbf{R}^n \mid \|x - x_0\|_2 \leq r\} \\ &= \{x \in \mathbf{R}^n \mid (x - x_0)^T(x - x_0) \leq r^2\}, \end{aligned}$$

where $r > 0$ is the radius and $x_0 \in \mathbf{R}^n$ is the center. The norm $\|\cdot\|_2$ is the *Euclidean norm*, *i.e.*, $\|u\|_2 = (u^T u)^{1/2}$. The ball $\mathcal{B}(x_0, r)$ consists of all points in $\mathbf{R}^n$ whose (Euclidean) distance from the center x_0 is at most r.

Euclidean balls are sometimes represented in the form

$$\mathcal{B}(x_0, r) = \{x_0 + ru \mid \|u\|_2 \leq 1\}.$$

Figure 2.9 (middle) shows the Euclidean ball centered at the origin with radius one (*i.e.*, the unit Euclidean ball) in $\mathbf{R}^2$.

Positive semidefinite cones

The set of $n \times n$ symmetric positive semidefinite matrices, which we denote as $\mathbf{S}^n_+$, is a convex cone: If $X_1, X_2 \in \mathbf{S}^n_+$ and $\theta_1, \theta_2 \geq 0$, then for any $z \in \mathbf{R}^n$, we have

$$z^T(\theta_1 X_1 + \theta_2 X_2)z = \theta_1 z^T X_1 z + \theta_2 z^T X_2 z \geq 0,$$

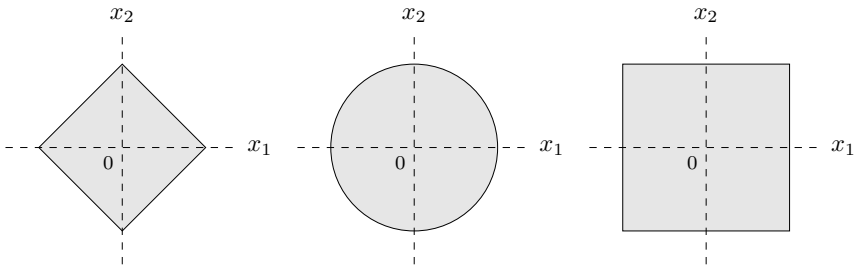

Figure 2.9 Examples of the unit norm balls in $\mathbf{R}^2$ for the norms: $\|\cdot\|_1$ (*left*), $\|\cdot\|_2$ (*middle*), and $\|\cdot\|_\infty$ (*right*). Each ball intersects the axes at $(0,1)$, $(1,0)$, $(0,-1)$, and $(-1,0)$.

where the inequality follows from the definition of positive semidefiniteness. This shows that the conic combination $\theta_1 X_1 + \theta_2 X_2$ is also a symmetric positive semidefinite matrix.

Similar results hold for the set of symmetric positive definite matrices $\mathbf{S}^n_{++}$, which is the interior of $\mathbf{S}^n_+$.

Example 2.5 *Positive semidefinite cone in* $\mathbf{S}^2$. Consider symmetric positive semidefinite matrices in $\mathbf{S}^2$ of the form

$$X = \begin{bmatrix} x & y \\ y & z \end{bmatrix} \in \mathbf{S}^2_+, \tag{2.6}$$

then we have

$$x \geq 0, \quad z \geq 0, \quad xz \geq y^2.$$

We can therefore plot the boundary of the cone $\mathbf{S}^2_+$ in $\mathbf{R}^3$ by transforming X into the vector (x, y, z), which is shown in figure 2.10.

2.2 Algebra of convex sets

This section describes some useful algebraic operations that preserve convexity of sets. These operations, together with the examples described in §2.1.3, form a calculus of convex sets that is useful for determining convexity of sets, or constructing new convex sets from others.

2.2.1 Intersection

Convexity of sets is preserved under intersection: If C_1 and C_2 are convex sets, then so is their intersection $C_1 \cap C_2$. This property can be generalized to the intersection of an arbitrary (possibly infinite) number of convex sets: If $\{C_i\}_{i\in\mathcal{I}}$ is a collection of convex sets indexed by $\mathcal{I}$, then their intersection $\bigcap_{i\in\mathcal{I}} C_i$ is also convex. The same property holds for affine sets and convex cones.

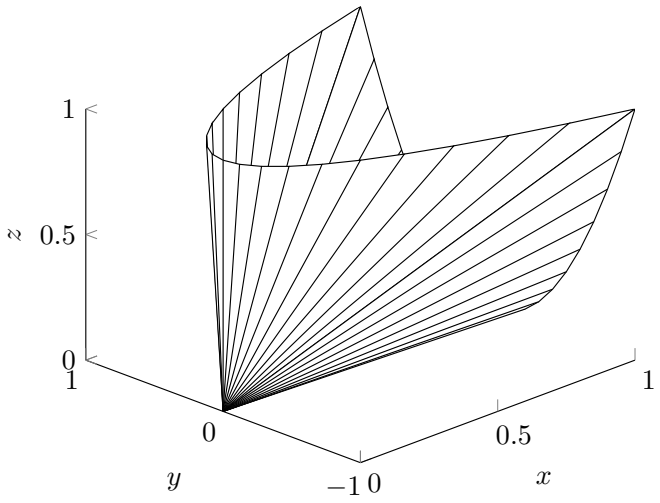

Figure 2.10 Boundary of the positive semidefinite cone $\mathbf{S}^2_+$ with points represented in coordinate $(x, y, z) \in \mathbf{R}^3$ given by (2.6).

Since convexity of sets is preserved under intersection, we can easily see that, for example, polyhedra are convex sets, since they are defined as the intersection of a finite number of halfspaces and hyperplanes, which are all convex.

Example 2.6 Let $X \in \mathbf{S}^n$ be a symmetric matrix. Recall that the matrix X is positive semidefinite if for all $z \in \mathbf{R}^n$ $(z \neq 0)$, we have

$$z^T X z \geq 0. \tag{2.7}$$

Note that for each fixed $z \in \mathbf{R}^n$ $(z \neq 0)$, the term $z^T X z$ is linear in X, and hence, the set of matrices $X \in \mathbf{S}^n$ that satisfy (2.7), *i.e.*,

$$\{X \in \mathbf{S}^n \mid z^T X z \geq 0\},$$

defines a halfspace in $\mathbf{S}^n$. Therefore, the positive semidefinite cone $\mathbf{S}^n_+$ can be expressed as the intersection of the halfspaces

$$\mathbf{S}^n_+ = \bigcap_{z \neq 0} \{X \in \mathbf{S}^n \mid z^T X z \geq 0\},$$

which is hence convex.

Example 2.7 The unit Euclidean ball in $\mathbf{R}^2$ can be expressed as the intersection of infinitely many halfspaces as

$$\mathcal{B} = \bigcap_{t \in [0, 2\pi)} \{x \in \mathbf{R}^2 \mid x_1 \cos t + x_2 \sin t \leq 1\},$$

and is hence convex. Geometrically, the boundary of each halfspace in the intersection corresponds to a line tangent to the unit Euclidean ball. This is illustrated in figure 2.11.

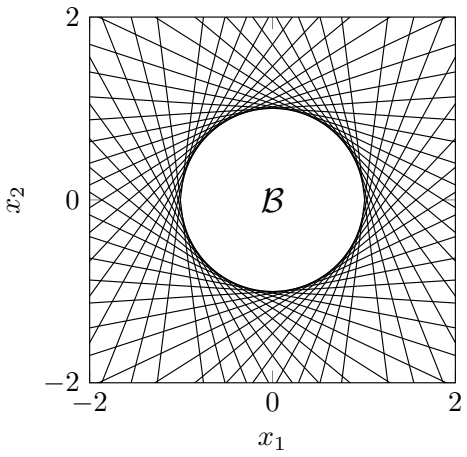

Figure 2.11 The unit Euclidean ball $\mathcal{B} \subseteq \mathbf{R}^2$ (white area in the middle) can be expressed as the intersection of infinitely many halfspaces. The boundary of 48 of these halfspaces are shown in lines.

In fact, such property can be generalized to any convex sets: Every (closed) convex set can be expressed as the intersection of (usually infinite) halfspaces that contain it; see exercise 2.5.

2.2.2 Cartesian product

The Cartesian product of convex sets is convex: If $C \subseteq \mathbf{R}^m$ and $D \subseteq \mathbf{R}^n$ are convex sets, then their Cartesian product

$$C \times D = \{(x, y) \mid x \in C,\ y \in D\} \subseteq \mathbf{R}^{m+n}$$

is convex. To show this, let $(x_1, y_1), (x_2, y_2) \in C \times D$ be two different points, where $x_1, x_2 \in C$ and $y_1, y_2 \in D$. According to the convexity of C and D, for any $\theta \in [0, 1]$, we have

$$\theta x_1 + (1-\theta)x_2 \in C \quad \text{and} \quad \theta y_1 + (1-\theta)y_2 \in D.$$

Hence, for any $\theta \in [0, 1]$, we have

$$\theta(x_1, y_1) + (1-\theta)(x_2, y_2) = (\theta x_1 + (1-\theta)x_2,\ \theta y_1 + (1-\theta)y_2) \in C \times D,$$

which shows that $C \times D$ is convex.

2.2.3 Affine transformation

An *affine transformation* is defined as a linear mapping plus a constant, *i.e.*, a function $f : \mathbf{R}^n \to \mathbf{R}^m$ is *affine* if it has the form

$$f(x) = Ax + b,$$

where $A \in \mathbf{R}^{m\times n}$ and $b \in \mathbf{R}^m$.

Suppose $C \subseteq \mathbf{R}^n$ is a convex set. If $f\colon \mathbf{R}^n \to \mathbf{R}^m$ is an affine function, then the *image* of C under f, given by

$$f(C) = \{f(x) \mid x \in C\} \subseteq \mathbf{R}^m,$$

is convex. Similarly, if $h\colon \mathbf{R}^k \to \mathbf{R}^n$ is affine, then the *inverse image* of C under h, given by

$$h^{-1}(C) = \{x \in \mathbf{R}^k \mid h(x) \in C\},$$

is convex.

With this property, we can show that many basic set operations are convexity preserving, *e.g.*,

- *Scaling and translation.* If $C \subseteq \mathbf{R}^n$ is convex, then for any $\lambda \in \mathbf{R}$ and $v \in \mathbf{R}^n$, the set
$$\lambda C + v = \{\lambda x + v \mid x \in C\}$$
is convex.

- *Addition.* If $C_1, C_2 \subseteq \mathbf{R}^n$ are convex, then so is their *sum*
$$C_1 + C_2 = \{x_1 + x_2 \mid x_1 \in C_1,\ x_2 \in C_2\}.$$
To see this, we can express $C_1 + C_2$ as the image of the Cartesian product $C_1 \times C_2$ (which is convex) under the affine function $f\colon \mathbf{R}^{n\times n} \to \mathbf{R}^n$ given by $f(x_1, x_2) = x_1 + x_2$.

Example 2.8 *Solution set of linear (matrix) inequality.* Consider the solution set of a system of linear inequalities:

$$\{x \in \mathbf{R}^n \mid Ax \preceq b\} \tag{2.8}$$

where $A \in \mathbf{R}^{m\times n}$ and $b \in \mathbf{R}^m$. Using the definition of convex sets, it can be easily verified that the set given by (2.8) is convex. Another way to see this is to express (2.8) as the inverse image of the nonnegative orthant $\mathbf{R}^m_+$ (which is a convex cone) under the affine function $f\colon \mathbf{R}^n \to \mathbf{R}^m$ given by $f(x) = b - Ax$:

$$\{x \mid Ax \preceq b\} = \{x \mid b - Ax \succeq 0\} = \{x \mid f(x) \in \mathbf{R}^n_+\}.$$

This idea can be generalized to the solution set of a *linear matrix inequality* in $x \in \mathbf{R}^n$:

$$\{x \in \mathbf{R}^n \mid A(x) \preceq B\}, \tag{2.9}$$

where $A(x)$ is defined as

$$A(x) = x_1A_1 + \cdots + x_nA_n$$

and $A_1, \ldots, A_n, B \in \mathbf{S}^m$. To see that the set given by (2.9) is convex, let $f\colon \mathbf{R}^n \to \mathbf{S}^m$ be the affine function given by $f(x) = B - A(x)$. Then, we can express (2.9) as

$$\{x \mid A(x) \preceq B\} = \{x \mid B - A(x) \succeq 0\} = \{x \mid f(x) \in \mathbf{S}^m_+\},$$

which is the inverse image of the positive semidefinite cone $\mathbf{S}^m_+$ under the affine function f.

Example 2.9 An *ellipsoid* in $\mathbf{R}^n$ is the set with the form

$$\mathcal{E} = \{x \in \mathbf{R}^n \mid (x - x_0)^T P^{-1}(x - x_0) \leq 1\}, \tag{2.10}$$

where $x_0 \in \mathbf{R}^n$ is the center and $P \in \mathbf{S}^n_{++}$. Ellipsoids are convex sets, which can be seen by express (2.10) as

$$\{x \in \mathbf{R}^n \mid \|P^{-1/2}(x - x_0)\|_2 \leq 1\},$$

which is the inverse image of the unit ball $\{x \in \mathbf{R}^n \mid \|x\|_2 \leq 1\}$ under the affine function $f(x) = P^{-1/2}(x - x_0)$.

Another commonly used representation of ellipsoids is

$$\mathcal{E} = \{x_0 + Au \mid \|u\|_2 \leq 1\},$$

where $A \in \mathbf{R}^{n \times n}$ is nonsingular, which is the image of the unit ball $\{u \in \mathbf{R}^n \mid \|u\|_2 \leq 1\}$ under the affine function $f(u) = x_0 + Au$ (and hence convex).

2.3 Convex functions

2.3.1 Definition

There are several equivalent definitions of convex functions. Here we present two of the most commonly used ones.

Epigraph

Recall that the graph of a function $f : \mathbf{R}^n \to \mathbf{R}$ is the set

$$\{(x, f(x)) \mid x \in \mathbf{dom}\, f\} \subseteq \mathbf{R}^{n+1}.$$

Similarly, the *epigraph* of f is a subset of $\mathbf{R}^{n+1}$, defined as

$$\mathbf{epi}\, f = \{(x, t) \in \mathbf{R}^{n+1} \mid f(x) \leq t,\ x \in \mathbf{dom}\, f\}.$$

This definition is illustrated in figure 2.12.

Example 2.10 The epigraph of affine functions are halfspaces: Let $f : \mathbf{R}^n \to \mathbf{R}$ be an affine function given by $f(x) = a^T x + b$, where $a \in \mathbf{R}^n$ $(a \neq 0)$ and $b \in \mathbf{R}$. Then, its epigraph is expressed as

$$\begin{aligned}\mathbf{epi}\, f &= \{(x, t) \in \mathbf{R}^{n+1} \mid a^T x + b \leq t\} \\ &= \left\{ \begin{bmatrix} x \\ t \end{bmatrix} \in \mathbf{R}^{n+1} \;\middle|\; \begin{bmatrix} -a^T & 1 \end{bmatrix} \begin{bmatrix} x \\ t \end{bmatrix} \geq b \right\},\end{aligned}$$

which is a halfspace in $\mathbf{R}^{n+1}$.

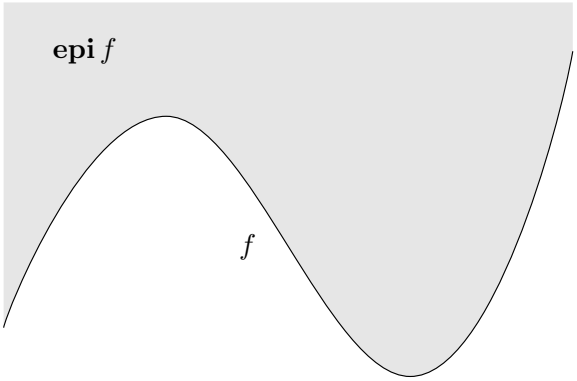

Figure 2.12 The epigraph of a function $f\colon \mathbf{R} \to \mathbf{R}$ (shown shaded) consists of all points (x, t) lying on or above the graph of f (lower boundary, shown thicker).

Epigraph allows us to define convex functions via convex sets: A function $f\colon \mathbf{R}^n \to \mathbf{R}$ is said to be *convex* if its epigraph $\mathbf{epi}\, f$ is a convex set in $\mathbf{R}^{n+1}$, and we say f is *concave* if $-f$ is convex, *i.e.*, if its *hypograph*

$$\mathbf{hypo}\, f = \{(x, t) \in \mathbf{R}^{n+1} \mid f(x) \geq t,\ x \in \mathbf{dom}\, f\}$$

is a convex set. An affine (and therefore, linear) function has both convex epigraph and hypograph, and is hence both convex and concave.

Jensen's inequality

An equivalent (under some technical conditions) definition of convex functions is given by the following inequality: A function $f\colon \mathbf{R}^n \to \mathbf{R}$ is convex if $\mathbf{dom}\, f$ is a convex set, and for all $x, y \in \mathbf{dom}\, f$ and $\theta \in [0, 1]$, we have

$$f(\theta x + (1 - \theta)y) \leq \theta f(x) + (1 - \theta)f(y). \tag{2.11}$$

If strict inequality holds in (2.11) for all $x \neq y$ and $\theta \in (0, 1)$, then f is said to be *strictly convex*. The condition (2.11) is sometimes called *Jensen's inequality*. Geometrically, it means that the line segment between the points $(x, f(x))$ and $(y, f(y))$ on the graph of f lies above the graph. This interpretation is illustrated in figure 2.13.

Jensen's inequality (2.11) can be generalized to more than two points: If a function f is convex, then for any points $x_1, \ldots, x_k \in \mathbf{dom}\, f$ and $\theta \in \mathbf{R}^k$ with $\mathbf{1}^T\theta = 1$ and $\theta \succeq 0$, we have

$$f(\theta_1 x_1 + \cdots + \theta_k x_k) \leq \theta_1 f(x_1) + \cdots + \theta_k f(x_k).$$

Jensen's inequality also extends to infinite sums, integrals, and expected values. For example, suppose $S \subseteq \mathbf{dom}\, f$, and let $p\colon S \to \mathbf{R}$ be a function satisfying

$$\int_S p(x)\, dx = 1 \quad \text{and} \quad p(x) \geq 0 \text{ for all } x \in S,$$

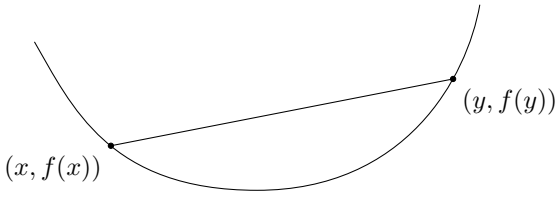

Figure 2.13 Graph of a convex function. The line segment between any two points on the graph lies above the graph.

then we have

$$f\left(\int_S p(x)x\ dx\right) \leq \int_S p(x)f(x)\ dx,$$

if all integrals exist. In the most general case, suppose x is a random variable with $\mathbf{prob}(x \in \mathbf{dom}\, f) = 1$, then we have

$$f(\mathbf{E}\, x) \leq \mathbf{E}\, f(x),$$

if all expectations exist. To recover the basic inequality (2.11) from the general form above, we may take the random variable $x \in \{x_1, x_2\} \subseteq \mathbf{dom}\, f$ with

$$\mathbf{prob}(x = x_1) = \theta \quad \text{and} \quad \mathbf{prob}(x = x_2) = 1 - \theta$$

for some $\theta \in [0, 1]$, then we have

$$f(\mathbf{E}\, x) = f(\theta x_1 + (1-\theta)x_2) \leq \mathbf{E}\, f(x) = \theta f(x_1) + (1-\theta)f(x_2).$$

2.3.2 Basic properties

Sublevel sets

The *α-sublevel set* of a function $f : \mathbf{R}^n \to \mathbf{R}$ is the set

$$C_\alpha = \{x \in \mathbf{dom}\, f \mid f(x) \leq \alpha\}.$$

If f is a convex function, then its sublevel sets are convex sets for any $\alpha \in \mathbf{R}$. To see this, let $x_1, x_2 \in C_\alpha$ be two different points, then for any $\theta \in [0, 1]$, we have

$$f(\theta x_1 + (1-\theta)x_2) \leq \theta f(x_1) + (1-\theta)f(x_2) \leq \theta\alpha + (1-\theta)\alpha = \alpha,$$

where the first inequality is from Jensen's inequality (2.11) and the second inequalities follow from the definition of C_α. This shows that the line segment $\theta x_1 + (1-\theta)x_2 \in C_\alpha$, and hence C_α is convex.

Remark 2.1 Note that the converse of the above property regarding sublevel sets of convex functions is not true: A function can have all its sublevel sets convex, but not be a convex function. For example, the sublevel sets of the function $f(x) = \log x$ can be expressed as the interval $(0, e^\alpha]$ for any $\alpha \in \mathbf{R}$, which are all convex sets, but f is not a convex function.

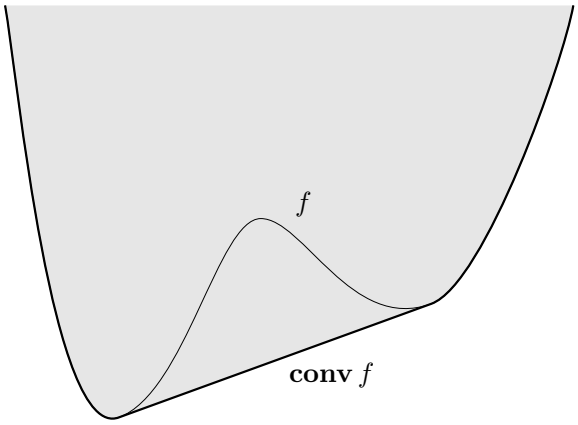

Figure 2.14 Convex envelope of the nonconvex function f is shown thicker. The epigraph of $\mathbf{conv}\, f$ is the convex hull of the epigraph of f (shown shaded).

Convex envelope

Recall that the convex hull of a set is the smallest convex set that contains it. We can also define an analogous concept for functions. First, note that the idea of epigraph provides an option to construct convex functions from convex sets: Suppose F is a convex set in $\mathbf{R}^{n+1}$, then the function $f\colon \mathbf{R}^n \to \mathbf{R}$ defined by

$$f(x) = \inf\{t \in \mathbf{R} \mid (x,t) \in F\} \tag{2.12}$$

is a convex function with epigraph $\mathbf{epi}\, f = F$. Let $g\colon \mathbf{R}^n \to \mathbf{R}$ be a (possibly nonconvex) function, then the function $f\colon \mathbf{R}^n \to \mathbf{R}$ obtained from the operation (2.12) with

$$F = \mathbf{conv}\, \mathbf{epi}\, g$$

is called the *convex envelope* of g, which we denote as $\mathbf{conv}\, g$ and is the largest convex function that is a *global underestimator* of g, *i.e.*,

$$f(x) = \sup\left\{ h(x) \;\middle|\; \begin{array}{l} h \text{ is convex} \\ h(z) \le g(z) \text{ for all } z \in \mathbf{dom}\, g \end{array} \right\}.$$

The idea of convex envelope is illustrated in figure 2.14.

Example 2.11 *Some important convex envelopes.* The *cardinality function* of a vector $x \in \mathbf{R}^n$, denoted as $\mathbf{card}\, x$, is defined as the number of nonzero entries in x. For scalar $x \in \mathbf{R}$, we have

$$\mathbf{card}\, x = \begin{cases} 0, & x = 0 \\ 1, & x \neq 0. \end{cases}$$

It is easily seen that the cardinality function is not convex. The convex envelope of the cardinality function on the interval $[-1, 1]$ is the absolute value function $|x|$. This idea is illustrated in figure 2.15.

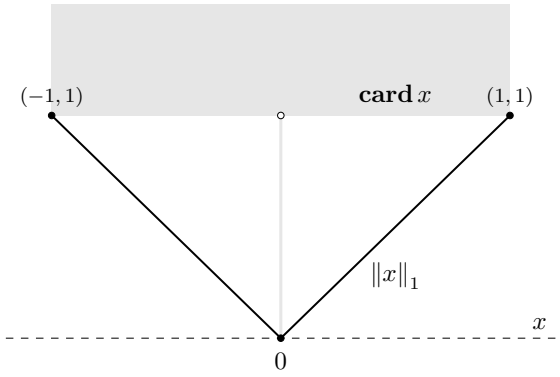

Figure 2.15 The cardinality function **card** x on the interval $[-1, 1]$. Its graph is shown as the thinner line and the dots; its epigraph is shown shaded. Its convex envelope (on the interval $[-1, 1]$) is the ℓ_1-norm $\|x\|_1 = |x|$ (shown thicker).

This property the absolute value function can be generalized to the ℓ_1-norm of vectors in $\mathbf{R}^n$: The convex envelope of the cardinality function on the unit ℓ_∞-norm ball $\{x \in \mathbf{R}^n \mid \|x\|_\infty \leq 1\}$, where $\|x\|_\infty = \max\{|x_1|, \ldots, |x_n|\}$, is the ℓ_1-norm $\|x\|_1 = |x_1| + \cdots + |x_n|$.

We have similar results for matrices: The *rank* function of a matrix $X \in \mathbf{R}^{m\times n}$, denoted as **rank** X, corresponds to the number of nonzero singular values of X. The convex envelope of the rank function on the unit *spectral norm* ball $\{X \in \mathbf{R}^{m\times n} \mid \|X\|_2 \leq 1\}$, where $\|X\|_2 = \sigma_{\max}(X)$, *i.e.*, the maximum singular value, is the *nuclear norm* $\|X\|_* = \sum_i \sigma_i(X)$, where $\sigma_i(X)$ are the singular values of X (see §A.3.2, page 354).

First-order conditions

If a function f is differentiable, then f is convex if and only if **dom** f is convex and for all $x, y \in$ **dom** f, we have

$$f(y) \geq f(x) + \nabla f(x)^T (y - x). \tag{2.13}$$

In other words, the first-order Taylor approximation of a differentiable convex function f at any point $x \in$ **dom** f is a global underestimator of f. This property is illustrated in figure 2.16. The converse is also true: If the first-order Taylor approximation of a differentiable function f at any point $x \in$ **dom** f is a global underestimator of f, then f is convex.

The first-order condition (2.13) can be used to define strict convexity: If strict inequality holds in (2.13) for all $x, y \in$ **dom** f and $x \neq y$, then f is strictly convex. The converse is also true.

Similarly, we have the following first-order condition for concave functions: A differentiable function f is concave if and only if **dom** f is convex and for all $x, y \in$

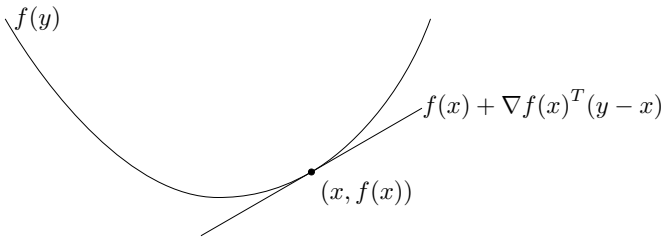

Figure 2.16 The first-order Taylor approximation of a convex function f at any point $x \in \mathbf{dom}\, f$ is a global underestimator of f.

$\mathbf{dom}\, f$, we have

$$f(y) \leq f(x) + \nabla f(x)^T(y - x). \tag{2.14}$$

Example 2.12 *Affine functions.* The first-order condition can be used to verify the convexity of affine functions: Let $f: \mathbf{R}^n \to \mathbf{R}$ be an affine function given by $f(x) = a^T x + b$ ($a \neq 0$). Then, for any $x, y \in \mathbf{R}^n$, we have

$$f(y) = a^T y + b = a^T x + b + a^T(y - x) = f(x) + \nabla f(x)^T(y - x),$$

where $\nabla f(x) = a$. This shows that affine functions satisfy the first-order conditions (2.13) and (2.14) with equality, and are hence both convex and concave.

Second-order conditions

If a function f is twice differentiable, then f is convex if and only if $\mathbf{dom}\, f$ is convex and for all $x \in \mathbf{dom}\, f$, we have

$$\nabla^2 f(x) \succeq 0, \tag{2.15}$$

i.e., the Hessian of f at every point in its domain is positive semidefinite. If f is defined on $\mathbf{R}$, then the condition (2.15) reduces to $f''(x) \geq 0$ for all $x \in \mathbf{dom}\, f$. Geometrically, this means that the graph of f has nonnegative (*i.e.*, upward) curvature at every point in its domain.

The second-order condition (2.15) can also be used to define strict convexity: If a function f satisfies $\nabla^2 f(x) \succ 0$ for all $x \in \mathbf{dom}\, f$, then f is strictly convex. Note that different from the first-order condition, the converse is *not* true: For example, the function $f(x) = x^4$ is strictly convex, but its second derivative at $x = 0$ is zero.

For concave functions, we have the following second-order condition: A twice differentiable function f is concave if and only if $\mathbf{dom}\, f$ is convex and for all $x \in \mathbf{dom}\, f$, we have

$$\nabla^2 f(x) \preceq 0.$$

Example 2.13 *Quadratic functions.* Let $f: \mathbf{R}^n \to \mathbf{R}$ be a quadratic function given by

$$f(x) = (1/2)x^T P x + q^T x + r,$$

where $P \in \mathbf{S}^n$, $q \in \mathbf{R}^n$, and $r \in \mathbf{R}$. Then, we have $\nabla^2 f(x) = P$ for all $x \in \mathbf{R}^n$. Therefore, according to the second-order condition (2.15), the function f is convex if and only if $P \succeq 0$, and is concave if and only if $P \preceq 0$.

For quadratic functions, the second-order condition is, in fact, the necessary and sufficient condition for strict convexity: The function f is strictly convex if and only if $P \succ 0$, and is strictly concave if and only if $P \prec 0$.

2.3.3 Examples

In the text above we have already seen that affine (and linear) functions are both convex and concave, and quadratic functions are convex (concave) if their Hessian are positive (negative) semidefinite. Now we present more important examples of convex and concave functions (whose convexity might not be obvious).

Real-valued functions

The convexity of the following functions on $\mathbf{R}$ follows directly from the Jensen's inequality (2.11) or the second-order condition (2.15):

- *Powers.* The function $f(x) = x^p$ is convex on $\mathbf{R}_{++}$ if $p \leq 0$ or $p \geq 1$, and is concave on $\mathbf{R}_{++}$ if $0 \leq p \leq 1$, which can be easily verified from the sign of its second-derivative $f''(x) = p(p-1)x^{p-2}$ for all $x \in \mathbf{R}_{++}$.
- *Exponential.* The function $f(x) = e^x$ is convex on $\mathbf{R}$ since $f''(x) = e^x > 0$ for all $x \in \mathbf{R}$.
- *Logarithm.* The function $f(x) = \log x$ is concave on $\mathbf{R}_{++}$ since $f''(x) = -1/x^2 < 0$ for all $x \in \mathbf{R}_{++}$.
- *Negative entropy.* The function $f(x) = x \log x$ is convex on $\mathbf{R}_{++}$ since $f''(x) = 1/x > 0$ for all $x \in \mathbf{R}_{++}$.

Vector-valued functions

We also have the following examples on $\mathbf{R}^n$:

- *Norms.* Every norm $\|\cdot\|$ on $\mathbf{R}^n$ is convex: For any $x, y \in \mathbf{R}^n$ and $\theta \in [0,1]$, we have

$$\|\theta x + (1-\theta)y\| \leq \|\theta x\| + \|(1-\theta)y\| = \theta\|x\| + (1-\theta)\|y\|,$$

  where the first inequality is from the triangle inequality and the second equality is from the homogeneity of norms.

- *Max function.* The function $f(x) = \max\{x_1, \ldots, x_n\}$ is convex on $\mathbf{R}^n$: For any $x, y \in \mathbf{R}^n$ and $\theta \in [0,1]$, we have

$$\max_i(\theta x_i + (1-\theta)y_i) \leq \theta \max_i x_i + (1-\theta)\max_i y_i,$$

  where the left-hand side equals to $f(\theta x + (1-\theta)y)$ and the right-hand side equals to $\theta f(x) + (1-\theta)f(y)$. The graph of the max function on $\mathbf{R}^2$ is shown in figure 2.17.

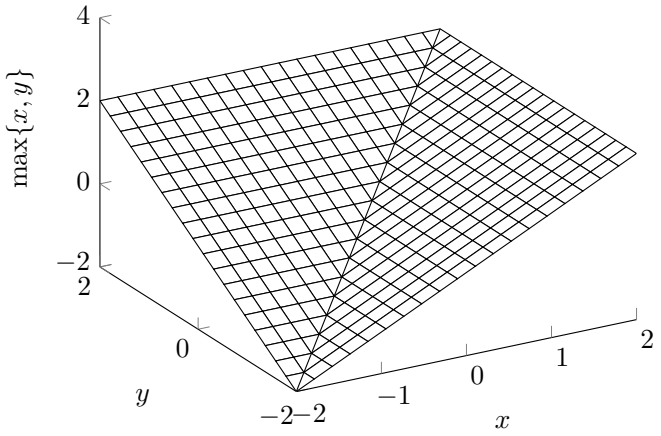

Figure 2.17 Graph of the max function $f(x, y) = \max\{x, y\}$ on $\mathbf{R}^2$.

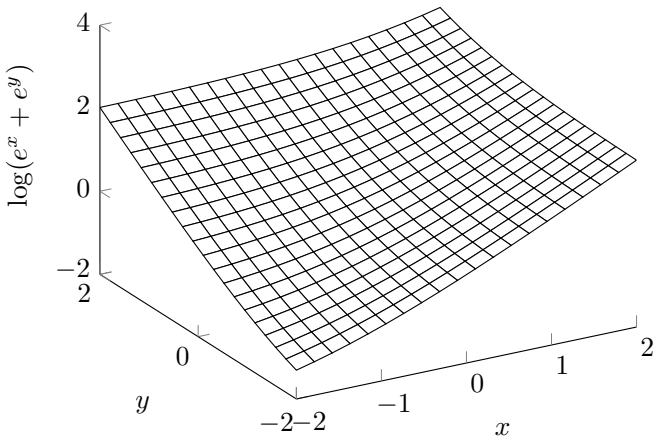

Figure 2.18 Graph of the log-sum-exp function $f(x, y) = \log(e^x + e^y)$ on $\mathbf{R}^2$.

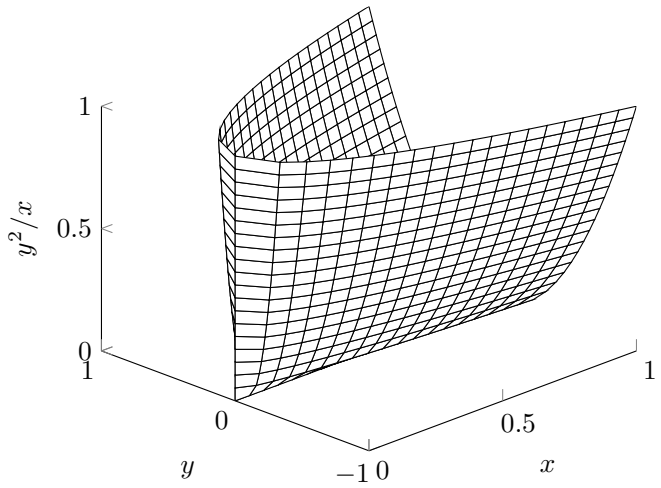

Figure 2.19 Graph of $f(x, y) = y^2/x$ on $\mathbf{R}_{++} \times \mathbf{R}$.

- *Log-sum-exp.* The function $f(x) = \log \sum_{i=1}^n \exp x_i$ is convex on $\mathbf{R}^n$ (which is not obvious, see §B.1). This function is often used as a smooth (*i.e.*, differentiable) approximation to the max function (figure 2.18).

- *Quadratic-over-linear.* The function $f(x, y) = y^2/x$ is convex on $\mathbf{R}_{++} \times \mathbf{R} = \{(x, y) \in \mathbf{R}^2 \mid x > 0\}$. In fact, it is the boundary of the positive semidefinite cone $\mathbf{S}_+^2$ (figure 2.19). (See also example 2.5.) To show this, note that for any $(x, y) \in \mathbf{R}_{++} \times \mathbf{R}$, we have

$$\nabla^2 f(x, y) = \frac{2}{x^3} \begin{bmatrix} y^2 & -xy \\ -xy & x^2 \end{bmatrix} = \frac{2}{x^3} \begin{bmatrix} y \\ -x \end{bmatrix} \begin{bmatrix} y \\ -x \end{bmatrix}^T \succeq 0.$$

- *Log-determinant.* The function $f(X) = \log \det X$ is concave on the positive definite cone $\mathbf{S}_{++}^n$, which can be interpreted as an analogous of the logarithm function for positive scalars, but for positive definite matrices.

- *Relative entropy.* The relative entropy between two positive vectors $x, y \in \mathbf{R}_{++}^n$ is given by

$$f(x, y) = \sum_{i=1}^n x_i \log(x_i/y_i),$$

  which is convex on $\mathbf{R}_{++}^n \times \mathbf{R}_{++}^n$; see exercise 2.14.

- *Kullback-Leibler divergence.* The Kullback-Leibler (KL) divergence between two positive vectors $x, y \in \mathbf{R}_{++}^n$ is given by

$$D_{\mathrm{kl}}(x, y) = \sum_{i=1}^n (x_i \log(x_i/y_i) - x_i + y_i), \tag{2.16}$$

which is convex in (x, y). The KL divergence satisfies $D_{\mathrm{kl}}(x, y) \geq 0$, and $D_{\mathrm{kl}}(x, y) = 0$ if and only if $x = y$, and so can be used as a measure of deviation between two positive vectors. Note that when both x and y are probability distributions (*i.e.*, $x, y \succeq 0$ and $\mathbf{1}^T x = \mathbf{1}^T y = 1$), the KL divergence is the same as the relative entropy.

Probability distributions

The *logarithm* of many probability density functions are concave, for example:

- *Multivariate normal distribution:*

$$f(x) = \frac{1}{\sqrt{(2\pi)^n \det \Sigma}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right),$$

  where $\mu \in \mathbf{R}^n$ is the mean and $\Sigma \in \mathbf{S}^n_{++}$ is the covariance matrix.

- *Uniform distribution:*

$$f(x) = \begin{cases} 1/\alpha, & x \in C \\ 0, & \text{otherwise,} \end{cases}$$

  where $C \subseteq \mathbf{R}^n$ is a convex set and $\alpha > 0$ is its volume. The logarithm of f is then given by

$$\log f(x) = \begin{cases} -\log \alpha, & x \in C \\ -\infty, & \text{otherwise,} \end{cases}$$

  which is concave.

The *logarithm* of some cumulative distribution functions (on $\mathbf{R}$) are also concave:

- *Gaussian cumulative distribution function:*

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-t^2/2}\, dt.$$

- *Logistic cumulative distribution function:*

$$F(x) = \frac{1}{1+e^{-x}}.$$

A function (or distribution) whose logarithm is concave is often called a *log-concave function*; see also exercise 2.11.

2.4 Functional operations

Similar to convex sets, many basic operations on functions preserve convexity. These operations, together with the examples presented in §2.3.3, allows us to easily determine the convexity of functions, or construct new convex functions from the others.

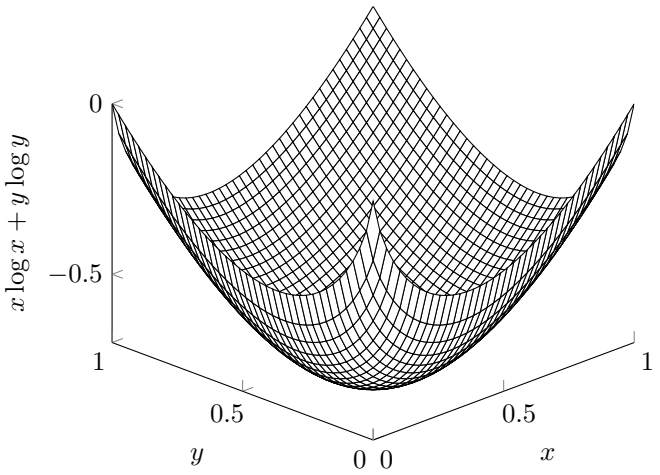

Figure 2.20 Graph of the negative entropy function $f(x, y) = x \log x + y \log y$ defined on $\mathbf{R}^2_{++}$.

2.4.1 Nonnegative weighted sums

We can easily verify that if $f_1, \ldots, f_k$ are convex functions and $w_1, \ldots, w_k \geq 0$, then the function

$$f = w_1 f_1 + \cdots + w_k f_k,$$

is also convex. This suggests that the set of convex functions is, in fact, a convex cone.

Example 2.14 *Negative entropy.* The negative entropy function on $\mathbf{R}^n_{++}$ is defined as:

$$f(x) = \sum_{i=1}^{n} x_i \log x_i,$$

which is convex since it is the sum of the convex functions $x_i \log x_i$ ($x_i \in \mathbf{R}_{++}$) for $i = 1, \ldots, n$. Graph of the negative entropy function on $\mathbf{R}^2_{++}$ is shown in figure 2.20.

Again, this property generalizes to infinite sums and integrals. For example, if $f(x, y)$ is convex in x for each fixed $y \in S$, and $w(y) \geq 0$ for all $y \in S$, then the function

$$g(x) = \int_S w(y) f(x, y)\, dy$$

is convex in x (provided the integral exists).

2.4.2 Pointwise maximum

Suppose $f_1, \ldots, f_k$ are convex functions, then the function

$$f(x) = \max\{f_1(x), \ldots, f_k(x)\}$$

with $\mathbf{dom}\, f = \bigcap_{i=1}^k \mathbf{dom}\, f_i$ is also convex. To see this, let $x_1, x_2 \in \mathbf{dom}\, f$ be two different points, then for any $\theta \in [0,1]$, we have

$$\begin{aligned} f(\theta x_1 + (1-\theta)x_2) &= \max_{i=1,\ldots,k} f_i(\theta x_1 + (1-\theta)x_2) \\ &\leq \max_{i=1,\ldots,k} (\theta f_i(x_1) + (1-\theta) f_i(x_2)) \\ &\leq \theta \max_{i=1,\ldots,k} f_i(x_1) + (1-\theta) \max_{i=1,\ldots,k} f_i(x_2) \\ &= \theta f(x_1) + (1-\theta) f(x_2), \end{aligned}$$

where the first inequality follows from the convexity of each f_i. We can also show this property using epigraphs: The epigraph of f can be expressed as the intersection of the epigraphs of f_i for $i = 1, \ldots, k$:

$$\mathbf{epi}\, f = \bigcap_{i=1}^k \mathbf{epi}\, f_i.$$

Since the intersection of convex sets is also convex, $\mathbf{epi}\, f$ is a convex set, and hence f is a convex function.

Similarly, for concave functions, the pointwise minimum of a set of concave functions is concave. Note that affine functions are both convex and concave, so the pointwise maximum of a set of affine functions is convex and the pointwise minimum of a set of affine functions is concave.

The pointwise maximum property generalizes to an infinite set of convex functions: Suppose $\{f_i\}_{i \in \mathcal{I}}$ is a collection of convex functions indexed by the (possibly infinite) set $\mathcal{I}$, then the function

$$f(x) = \sup_{i \in \mathcal{I}} f_i(x)$$

with

$$\mathbf{dom}\, f = \left\{ x \in \bigcap_{i \in \mathcal{I}} \mathbf{dom}\, f_i \;\middle|\; \sup_{i \in \mathcal{I}} f_i(x) < \infty \right\}$$

is also convex. It follows directly from the fact that $\mathbf{epi}\, f$ can be expressed as the intersection of the epigraphs of f_i for all $i \in \mathcal{I}$, which are all convex sets.

Example 2.15 *Piecewise linear functions.* A function $f : \mathbf{R}^n \to \mathbf{R}$ with the form

$$f(x) = \max_{i=1,\ldots,k} (a_i^T x + b_i),$$

where $a_i \in \mathbf{R}^n$ $(a_i \neq 0)$ and $b_i \in \mathbf{R}$, is called a piecewise linear function (or really, piecewise affine; although the first name is more commonly used). A piecewise linear function is convex as it is the pointwise maximum of the affine functions $a_i^T x + b_i$ over $i = 1, \ldots, k$. The epigraph of a piecewise linear function is a polyhedron, since it is the intersection of the halfspaces defined by the affine functions. Example of a piecewise linear function on $\mathbf{R}$ is shown in figure 2.21.

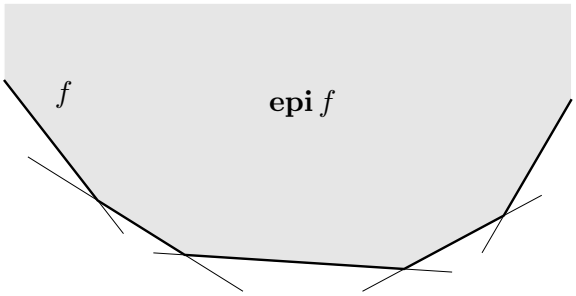

Figure 2.21 A piecewise linear function (shown thicker) from the maximum of five affine functions on $\mathbf{R}$. The epigraph of the piecewise linear function (shown shaded) is the intersection of the epigraphs of the five affine functions.

The converse (with slight extension) is also true: Any convex function $f\colon \mathbf{R}^n \to \mathbf{R}$ can be expressed as the pointwise supremum of all its affine underestimators (which are potentially infinite), *i.e.*,

$$f(x) = \sup\{a^T x + b \mid a^T y + b \le f(y) \text{ for all } y \in \mathbf{R}^n\}$$

(see exercise 2.8).

Example 2.16 *Sum of k largest entries.* Consider the function $f\colon \mathbf{R}^n \to \mathbf{R}$ given by

$$f(x) = \sum_{i=1}^{k} x_{[i]},$$

where $k \le n$ and $x_{[i]}$ denotes the ith largest entry of x, *i.e.*,

$$x_{[1]} \ge x_{[2]} \ge \cdots \ge x_{[n]}.$$

The function f is convex, since it can be expressed as the pointwise maximum of all possible sums of k different entries of x:

$$f(x) = \sum_{i=1}^{k} x_{[i]} = \max\{x_{i_1} + \cdots + x_{i_k} \mid 1 \le i_1 \le \cdots \le i_k \le n\},$$

which is the pointwise maximum of a set of linear functions.

2.4.3 Composition with affine function

Let $f\colon \mathbf{R}^n \to \mathbf{R}$ be a function, and let $A \in \mathbf{R}^{n\times m}$ and $b \in \mathbf{R}^n$. Then the function $g\colon \mathbf{R}^m \to \mathbf{R}$ defined as

$$g(x) = f(Ax + b)$$

with $\mathbf{dom}\, g = \{x \mid Ax + b \in \mathbf{dom}\, f\}$ has the same convexity as f, *i.e.*, if f is convex, so is g, and if f is concave, so is g. We can shown this directly via Jensen's

inequality (2.11). Suppose f is convex, and let $x_1, x_2 \in \mathbf{dom}\, g$ be two different points, then for any $\theta \in [0,1]$, we have

$$\begin{aligned} g(\theta x_1 + (1-\theta)x_2) &= f(A(\theta x_1 + (1-\theta)x_2) + b) \\ &= f(\theta(Ax_1 + b) + (1-\theta)(Ax_2 + b)) \\ &\leq \theta f(Ax_1 + b) + (1-\theta) f(Ax_2 + b) \\ &= \theta g(x_1) + (1-\theta) g(x_2), \end{aligned}$$

where the inequality follows from the convexity of f.

Example 2.17 *Least squares cost.* The least squares cost function $f: \mathbf{R}^n \to \mathbf{R}$ given by

$$f(x) = \|Ax - b\|_2^2$$

with $A \in \mathbf{R}^{m \times n}$ and $b \in \mathbf{R}^m$ is a convex function. To show this, first note that for all $u \in \mathbf{R}^m$, we have

$$\nabla^2 \|u\|_2^2 = 2I \succeq 0$$

($I \in \mathbf{R}^{m \times m}$ is the identity matrix), which shows that the Euclidean norm squared function $\|\cdot\|_2^2$ is a convex function. Hence, the least squares cost f can be expressed as the composition of the convex function $\|\cdot\|_2^2$ with the affine function $Ax - b$, and is therefore convex.

2.4.4 General composition

Given the functions $h: \mathbf{R}^k \to \mathbf{R}$ and $g_i: \mathbf{R}^n \to \mathbf{R}$ for $i = 1, \ldots, k$, we consider the function $f: \mathbf{R}^n \to \mathbf{R}$ defined as the composition

$$f(x) = h(g_1(x), \ldots, g_k(x))$$

with

$$\mathbf{dom}\, f = \left\{ x \in \mathbf{R}^n \;\middle|\; \begin{array}{l} x \in \mathbf{dom}\, g_i,\ i = 1, \ldots, k \\ (g_1(x), \ldots, g_k(x)) \in \mathbf{dom}\, h \end{array} \right\}.$$

The function f is convex if $\mathbf{dom}\, f$ is convex, the function h is convex, and one of the following conditions holds for each $i = 1, \ldots, k$:

$$\begin{array}{l} g_i \text{ is affine,} \\ h \text{ is nondecreasing in its } i\text{th argument, and } g_i \text{ is convex,} \\ h \text{ is nonincreasing in its } i\text{th argument, and } g_i \text{ is concave.} \end{array} \tag{2.17}$$

Similarly, the function f is concave if $\mathbf{dom}\, f$ is convex, the function h is concave, and one of the following conditions holds for each $i = 1, \ldots, k$:

$$\begin{array}{l} g_i \text{ is affine,} \\ h \text{ is nondecreasing in its } i\text{th argument, and } g_i \text{ is concave,} \\ h \text{ is nonincreasing in its } i\text{th argument, and } g_i \text{ is convex.} \end{array} \tag{2.18}$$

The composition rules (2.17) and (2.18) includes, as special cases, the composition with affine functions described above.

We can show the convexity condition (2.17) for scalar ($k = n = 1$) and differentiable functions by examining the second derivative of the composition $f = h \circ g$. Let $h\colon \mathbf{R} \to \mathbf{R}$ and $g\colon \mathbf{R} \to \mathbf{R}$ be twice differentiable functions, and $\mathbf{dom}\, h \circ g$ is convex, then we have

$$f''(x) = h''(g(x))g'(x)^2 + h'(g(x))g''(x).$$

Suppose h is convex and nondecreasing, and g is convex, then we have $h''(g(x)) \geq 0$, $h'(g(x)) \geq 0$ and $g''(x) \geq 0$ for all $x \in \mathbf{dom}\, f$. This shows that $f''(x) \geq 0$ for all $x \in \mathbf{dom}\, f$, and hence f is convex. Now suppose h is convex and nonincreasing, and g is concave, then we have $h''(g(x)) \geq 0$, $h'(g(x)) \leq 0$ and $g''(x) \leq 0$ for all $x \in \mathbf{dom}\, f$. This again shows that $f''(x) \geq 0$ for all $x \in \mathbf{dom}\, f$, and hence f is convex.

Convexity of many important functions on $\mathbf{R}$ can be shown using these composition rules.

Example 2.18 *Scalar composition.*

- The function $f(x) = |x|^p$ is convex if $p \geq 1$, since the function $h(u) = u^p$ is convex and nondecreasing on $\mathbf{R}_+$ for $p \geq 1$, and the function $g(x) = |x|$ is nonnegative and convex.
- The function $f(x) = 1/g(x)$ is convex if g is concave and positive, since the function $h(u) = 1/u$ is convex and nonincreasing on $\mathbf{R}_{++}$.
- The function $f(x) = \exp g(x)$ is convex if g is convex, since the exponential function is convex and nondecreasing.
- The function $f(x) = \log g(x)$ is concave if g is concave and positive, since the logarithm function is concave and nondecreasing on $\mathbf{R}_{++}$.

Similarly, we can use the composition rules to show the convexity of many functions on $\mathbf{R}^n$.

Example 2.19 *Vector composition.*

- The function $f(x) = \left(\sum_{i=1}^k g_i(x)^p\right)^{1/p}$ with $p \geq 1$ is convex if $g_1, \ldots, g_k$ is convex and nonnegative, since $h(u) = \left(\sum_{i=1}^k u_i^p\right)^{1/p}$ on $\mathbf{R}_+^k$ is convex and nondecreasing in u_i if $p \geq 1$. (In fact, it is the ℓ_p-norm of the nonnegative vector $u \in \mathbf{R}_+^k$; see example A.1.)
- The function $f(x) = \log \sum_{i=1}^k \exp g_i(x)$ is convex if $g_1, \ldots, g_k$ are convex, since the log-sum-exp function $h(u) = \log \sum_{i=1}^k \exp u_i$ on $\mathbf{R}^k$ is convex and nondecreasing in u_i.
- Suppose $g_1, \ldots, g_k\colon \mathbf{R}^n \to \mathbf{R}$ are convex functions, then the sum of r largest entries ($r \leq k$) of the vector $(g_1(x), \ldots, g_k(x))$ is convex, since the sum of r largest entries of a vector $u \in \mathbf{R}^k$, given by $h(u) = \sum_{i=1}^r u_{[i]}$, is convex and nondecreasing in each argument.

2.5 Convex optimization problems

2.5.1 Optimization problems

A *mathematical optimization problem*, or just *problem*, is denoted as

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \ldots, m \\ & h_i(x) = 0, \quad i = 1, \ldots, p, \end{array} \tag{2.19}$$

where $x \in \mathbf{R}^n$ is the *optimization variable*, or *decision variable*, or just *variable*, and $f_0 \colon \mathbf{R}^n \to \mathbf{R}$ is the *objective function* or *cost function*. The inequalities $f_i(x) \leq 0$, $i = 1, \ldots, m$, are the *inequality constraints*, and $f_1, \ldots, f_m \colon \mathbf{R}^n \to \mathbf{R}$ are the *inequality constraint functions*. The equalities $h_i(x) = 0$, $i = 1, \ldots, p$, are the *equality constraints*, and $h_1, \ldots, h_p \colon \mathbf{R}^n \to \mathbf{R}$ are the *equality constraint functions*. The goal of an optimization problem in the form (2.19) is to find an $x \in \mathbf{R}^n$ that minimizes $f_0(x)$ which satisfies the conditions $f_i(x) \leq 0$ for all $i = 1, \ldots, m$ and $h_i(x) = 0$ for all $i = 1, \ldots, p$.

The problem (2.19) is called a *constrained optimization problem* if $m > 0$ or $p > 0$. If $m = p = 0$, then the problem (2.19) is called an *unconstrained optimization problem*.

The *feasible set* of the problem (2.19) is defined as

$$\mathcal{D} = \left\{ x \in \bigcap_{i=0}^{m} \mathbf{dom}\, f_i \cap \bigcap_{i=1}^{p} \mathbf{dom}\, h_i \;\middle|\; \begin{array}{l} f_i(x) \leq 0, \; i = 1, \ldots, m \\ h_i(x) = 0, \; i = 1, \ldots, p \end{array} \right\}.$$

A point $x \in \mathcal{D}$ is called a *feasible point* of the problem (2.19). The problem (2.19) is *feasible* if its feasible set $\mathcal{D}$ is nonempty, and is otherwise *infeasible*.

The problem (2.19) is sometimes referred to as a *minimization problem*. An optimization problem that seeks to maximize some objective function is denoted as

$$\begin{array}{ll} \text{maximize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \ldots, m \\ & h_i(x) = 0, \quad i = 1, \ldots, p, \end{array} \tag{2.20}$$

and is called a *maximization problem*. A maximization problem can be converted to a minimization problem by minimizing $-f_0$.

A problem of the form

$$\begin{array}{ll} \text{find} & x \\ \text{subject to} & f_i(x) \geq 0, \quad i = 1, \ldots, m \\ & h_i(x) = 0, \quad i = 1, \ldots, p, \end{array} \tag{2.21}$$

is called a *feasibility problem*, where the goal is to examine whether there exists a point x that satisfies all the constraints, and if so, to find one.

Optimal value and points

The *optimal value* of the problem (2.19) is defined as

$$p^\star = \inf_{x\in\mathcal{D}} f_0(x) = \inf \left\{ f_0(x) \;\middle|\; \begin{array}{ll} x \in \bigcap_{i=0}^m \mathbf{dom}\, f_i \cap \bigcap_{i=1}^p \mathbf{dom}\, h_i \\ f_i(x) \le 0, & i = 1,\dots,m \\ h_i(x) = 0, & i = 1,\dots,p \end{array} \right\}.$$

Following from the standard convention, we have $p^\star = \infty$ if the problem is infeasible (since the infimum of an empty set is ∞), and $p^\star = -\infty$ if the problem is *unbounded below* (that is, for any $M \in \mathbf{R}$, there exists a feasible point x such that $f_0(x) < M$). All these terms can be defined similarly for maximization problems the the form (2.20).

A point $x^\star \in \mathcal{D}$ is called an *optimal point* (or *solution*) of the problem (2.19) if $f_0(x^\star) = p^\star$. If there exists at least one optimal point to the problem (2.19), then we say the optimal value is *attained* or *achieved*.

Example 2.20 *Optimal value and points.* Consider some unconstrained (minimization) problems with objective $f_0 \colon \mathbf{R} \to \mathbf{R}$ and $\mathbf{dom}\, f = \mathbf{R}_{++}$:

- Let $f_0(x) = 1/x$, then the optimal value is $p^\star = 0$, but it is not achieved since there is no feasible point $x \in \mathbf{R}_{++}$ such that $f_0(x) = 0$.
- Let $f_0(x) = -\log x$, then the optimal value is $p^\star = -\infty$, which means the problem is unbounded below.
- Let $f_0(x) = x \log x$, then the optimal value is $p^\star = -1/e$, which is achieved at the (unique) optimal point $x^\star = 1/e$.

A feasible point $x \in \mathcal{D}$ is *locally optimal* if there exists some $\epsilon > 0$ such that

$$f_0(x) = \inf\{f_0(z) \mid z \in \mathcal{D},\ \|z - x\|_2 \le \epsilon\}.$$

Roughly speaking, a locally optimal point is one that has the lowest objective value among all feasible points in its neighborhood. To distinguish between 'optimal' and 'locally optimal', the former is sometimes referred to as *globally optimal*.

Equivalent problems and relaxations

It is sometimes useful to transform an optimization problem into another one that is easier to analyze or solve. If from an optimal point of one problem, we can easily find an optimal point for another problem, and vice versa, then we say that the two problems are *equivalent*. Some basic operations that lead to equivalent problems include *change of variables*, *scaling and translating the objective*, etc. The following examples introduce two other useful transformations that lead to equivalent problems, which are not directly obvious.

Example 2.21 *Slack variables.* We can transform the inequality constraints of an optimization problem into equality constraints by introducing additional variables.

Noticing that in the problem (2.19), the conditions $f_i(x) \leq 0$ are satisfied if and only if there exists some $s \in \mathbf{R}^m$ and $s \succeq 0$ such that $f_i(x) + s_i = 0$, we can transform the problem (2.19) into

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & s \succeq 0 \\ & f_i(x) + s_i = 0, \quad i = 1, \ldots, m \\ & h_i(x) = 0, \quad i = 1, \ldots, p, \end{array} \tag{2.22}$$

where the variables are now $x \in \mathbf{R}^n$ and $s \in \mathbf{R}^m$. The new variable s_i is called the *slack variable* associated with the inequality constraint $f_i(x) \leq 0$. It is easy to see that the problems (2.19) and (2.22) are equivalent: If $(x^\star, s^\star)$ is an optimal point of the problem (2.22), then $x^\star$ is an optimal point of the problem (2.19), since $f_i(x^\star) = -s_i^\star \leq 0$ for all $i = 1, \ldots, m$; conversely, if $x^\star$ is an optimal point of the problem (2.19), then $(x^\star, s^\star)$ with $s_i^\star = -f_i(x^\star)$ for $i = 1, \ldots, m$ is an optimal point of the problem (2.22).

Example 2.22 *Epigraph form problems.* The equivalent *epigraph form* of the optimization problem (2.19) is given by

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & f_0(x) - t \leq 0 \\ & f_i(x) \leq 0, \quad i = 1, \ldots, m \\ & h_i(x) = 0, \quad i = 1, \ldots, p, \end{array} \tag{2.23}$$

where the variables are $x \in \mathbf{R}^n$ and $t \in \mathbf{R}$. It is easily seen that if $(x^\star, t^\star)$ is an optimal point of (2.23), which means the inequality $f_0(x^\star) \leq t^\star$ must hold with equality, then $x^\star$ is an optimal point of the problem (2.19); if $x^\star$ is an optimal point of (2.19), then $(x^\star, t^\star)$ with $t^\star = f_0(x^\star)$ is an optimal point of (2.23).

Geometrically, the epigraph form problem (2.23) can be interpreted as an optimization problem in the 'graph space' (x, t), *i.e.*, we want to find a point $(x, t) \in \mathbf{epi}\, f_0$ with the smallest t among all other points in $\mathbf{epi}\, f_0$, subject to some constraints on x. This interpretation is illustrated in figure 2.22.

If two optimization problems have the same objective function f_0, but different feasible sets $\mathcal{D}_1$ and $\mathcal{D}_2$ such that $\mathcal{D}_1 \subseteq \mathcal{D}_2$, then we say that the second problem is a *relaxation*, or a *relaxed problem*, of the first one. It is easy to see that the optimal value of a relaxed problem provides a lower bound for the optimal value of the original problem in minimization problems, and an upper bound in maximization problems. (The idea of relaxations will be discussed in much more detail in §6.5.)

Example 2.23 *Integer programming.* Consider the optimization problem

$$\begin{array}{ll} \text{minimize} & f_0(x_1, \ldots, x_m) \\ \text{subject to} & x_1, \ldots, x_m \in \{0, 1\}^n \\ & \mathbf{card}\, x_i = 1, \quad i = 1, \ldots, m \end{array} \tag{2.24}$$

with variables $x_1, \ldots, x_m \in \mathbf{R}^n$. The problem (2.24) is sometimes called a *integer program* since the variables x_i are constrained to take values in the discrete set $\{0, 1\}^n$.

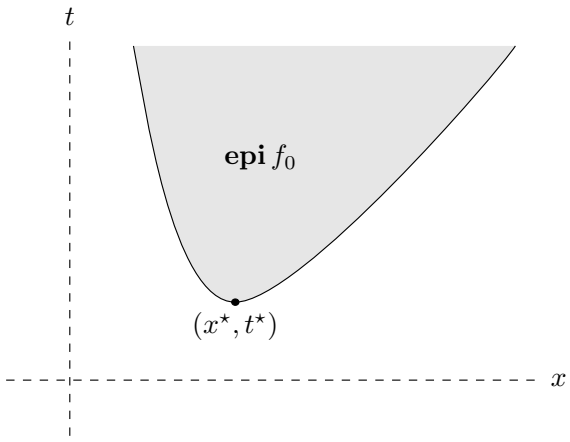

Figure 2.22 Geometric interpretation of the epigraph form problem (2.23). The goal is to find a point (x, t) in the epigraph of f_0 (the shaded region) with the smallest t among all other points in the epigraph.

We can interpret the constraints in (2.24) as requiring that $x_i \in \{e_1, \ldots, e_n\}$ for $i = 1, \ldots, m$, where e_i is the ith standard basis vector in $\mathbf{R}^n$. In other words, the problem (2.24) consists in selecting a group of points $x_1, \ldots, x_m$ from the vertices of the probability simplex in $\mathbf{R}^n$, such that $f_0(x_1, \ldots, x_m)$ is minimized. The problem (2.24) is in general very difficult to solve directly since we need to evaluate all n^m combinations of the possible values of $x_1, \ldots, x_m$. In practice, we often consider a relaxation of the problem (2.24) given by

$$\begin{array}{ll} \text{minimize} & f_0(x_1, \ldots, x_m) \\ \text{subject to} & x_1, \ldots, x_m \succeq 0 \\ & \mathbf{1}^T x_i = 1, \quad i = 1, \ldots, m, \end{array} \tag{2.25}$$

where the constraints now require that x_i lies *in* the probability simplex in $\mathbf{R}^n$ for $i = 1, \ldots, m$. It is easy to see that the feasible set of the relaxed problem (2.25) contains that of the original problem (2.24), since all vertices of a probability simplex also belong to the simplex.

We can interpret the problems (2.24) and (2.25) in the context of clustering: Suppose we have m data points that we want to assign to n different clusters, and the cost function f_0 measures some notion of total clustering error. The problem (2.24) consists in assigning each data point a label to a single cluster, such that the total clustering error is minimized. The relaxed problem (2.25), on the other hand, tries to assign each data point a probability distribution over all clusters that leads to the least expected total clustering error.

Specifying an optimization problem

For an optimization problem in the form (2.19), there is still the technical question of how to specify the objective and constraint functions. Most commonly, these

functions are expressed analytically, *i.e.*, consist of a number of arithmetic operations that involves the variables x and some parameters. For example, suppose the objective is some quadratic function in the form $f_0(x) = x^T Px + q^T x + r$ with $P \in \mathbf{S}^n$, $q \in \mathbf{R}^n$ and $r \in \mathbf{R}$. Then, to specify this objective, we need to provide the values of the coefficients P, q and r. We call all such coefficients required to specify an optimization problem the *problem parameters* or *problem data*.

Remark 2.2 Note that in the context of model fitting, we should carefully distinguish what exactly the word 'parameter' refers to: The *model parameters* are the variables to be optimized over in the fitting process, while the *problem parameters* are the coefficients that define the optimization problem itself. In most cases, it should be clear from the context or text which type of parameters is being referred to, but to avoid possible confusion, in this book, we will always use the term *problem data* when referring to the coefficients that define an optimization problem.

In the other cases, the objective and constraint functions may not be expressed analytically, but are given implicitly via some numerical procedure or algorithm, *i.e.*, via some *oracle* models. For example, in an oracle model of a function f, we do not know f explicitly, but can evaluate $f(x)$ (and usually its derivatives $\nabla f(x)$ and $\nabla^2 f(x)$) at any $x \in \mathbf{dom}\, f$, with some cost, *e.g.*, time.

As a matter of practice, there is not huge difference between specifying a problem via parameterized expressions or some oracle models, since most optimization algorithms only require the ability to evaluate the objective and constraint functions (and their derivatives) at different points, and do not need to know the explicit forms of these functions.

2.5.2 Convex optimization

A *convex optimization problem* (or *convex program*) is defined as

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \ldots, m \\ & Ax = b \end{array} \tag{2.26}$$

with variable $x \in \mathbf{R}^n$, where $f_0, f_1, \ldots, f_m \colon \mathbf{R}^n \to \mathbf{R}$ are convex, and $A \in \mathbf{R}^{p \times n}$, $b \in \mathbf{R}^p$. The problem (2.26) is a special case of the general optimization problem (2.19), where the objective and inequality constraint functions $f_0, f_1, \ldots, f_m$ are all convex and the equality constraint functions $h_1, \ldots, h_p$ are all affine. It follows immediately from this definition that the feasible set of a convex optimization problem is a convex set, since it is the intersection of the following convex sets:

- The domains $\mathbf{dom}\, f_i$, $i = 0, \ldots, m$, of the convex functions $f_0, f_1, \ldots, f_m$.
- The sublevel sets $\{x \mid f_i(x) \leq 0\}$ for $i = 1, \ldots, m$.
- The affine sets $\{x \mid Ax = b\}$.

Hence, roughly speaking, a convex optimization problem consists in minimizing a convex function over a convex set.

Note that the convex minimization problem (2.26) can always be transformed into the equivalent *concave maximization problem*:

$$\begin{array}{ll} \text{maximize} & -f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \ldots, m \\ & Ax = b, \end{array} \tag{2.27}$$

where the objective function is now concave, and vice versa. Hence, in general, when talking about convex optimization problems, we refer to problems in both the convex minimization form (2.26) and the concave maximization form (2.27).

Optimal points

One of the most important properties of convex optimization problems is that any locally optimal point is also globally optimal. To see this, let $\mathcal{D}$ be the feasible set of the problem (2.26), and let $x \in \mathcal{D}$ be a locally optimal point of (2.26) with some $\epsilon > 0$, then we have

$$f_0(x) = \inf\{f_0(z) \mid z \in \mathcal{D},\ \|z - x\|_2 \leq \epsilon\}. \tag{2.28}$$

Suppose there exists some point $y \in \mathcal{D}$ such that $f_0(y) < f_0(x)$, *i.e.*, the point x is *not* globally optimal, then we have $\|y - x\|_2 > \epsilon$, since otherwise it contradicts (2.28). Noticing that the feasible set $\mathcal{D}$ is convex, we have for any $\theta \in (0, 1)$,

$$z = \theta y + (1 - \theta)x \in \mathcal{D}.$$

Choosing

$$\theta = \frac{\epsilon}{2\|y - x\|_2} \in (0, 1),$$

then $z \in \mathcal{D}$ and $\|z - x\|_2 = \epsilon/2 < \epsilon$, and hence by (2.28), we have $f_0(x) \leq f_0(z)$. However, from the convexity of f_0, we have

$$f_0(z) \leq (1 - \theta)f_0(x) + \theta f_0(y) < f_0(x),$$

which is a contradiction. Therefore, the point x must be globally optimal.

Optimality conditions

When the convex optimization problem (2.26) has differentiable objective f_0, a necessary and sufficient condition for a feasible point $x \in \mathcal{D}$ to be optimal is given by

$$\nabla f_0(x)^T (y - x) \geq 0 \tag{2.29}$$

for all $y \in \mathcal{D}$. When $\nabla f_0(x) \neq 0$, the condition (2.29) has a nice geometric interpretation: The hyperplane with normal vector $-\nabla f_0(x)$ supports the feasible set $\mathcal{D}$ at the point x, *i.e.*, the feasible set $\mathcal{D}$ lies entirely in the halfspace

$$\left\{ z \in \mathbf{R}^n \;\middle|\; -\nabla f_0(x)^T z \leq -\nabla f_0(x)^T x \right\}.$$

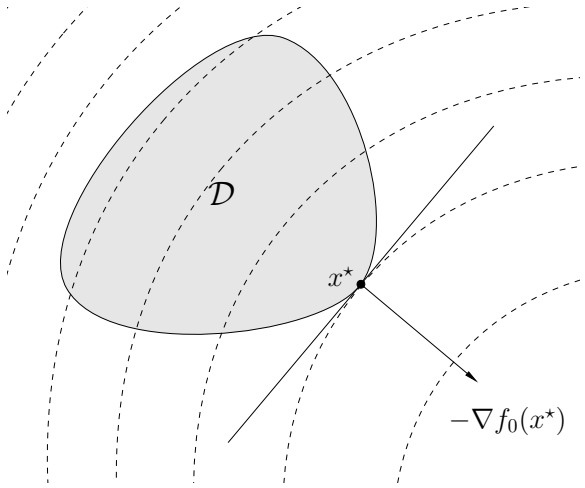

Figure 2.23 Geometric interpretation of the optimality condition (2.29) for convex optimization problems. The feasible set $\mathcal{D}$ is shown shaded, and the level curves of the objective function f_0 are shown as dashed lines. At the optimal point $x^\star$, the gradient $\nabla f_0(x^\star)$ defines a hyperplane (the solid line) that supports the feasible set $\mathcal{D}$.

This interpretation is illustrated in figure 2.23.

If the problem (2.26) is unconstrained, *i.e.*, $m = 0$ and $p = 0$, then the optimality condition (2.29) reduces to

$$\nabla f_0(x) = 0,$$

which is the well known first-order necessary and sufficient optimality condition for unconstrained convex optimization problems. To show this, first notice that the feasible set of an unconstrained problem is $\mathcal{D} = \mathbf{dom}\, f_0$. Suppose $x \in \mathbf{dom}\, f_0$ is optimal, then by the optimality condition (2.29), we have $\nabla f_0(x)^T(y - x) \geq 0$ for any $y \in \mathbf{dom}\, f_0$. Choosing $y = x - t\nabla f_0(x)$ with some $t > 0$ sufficiently small such that $y \in \mathbf{dom}\, f_0$, we have

$$\nabla f_0(x)^T(y - x) = -t\|\nabla f_0(x)\|_2^2 \geq 0,$$

which implies $\nabla f_0(x) = 0$. Conversely, if $\nabla f_0(x) = 0$, then for any $y \in \mathbf{dom}\, f_0$, we have $\nabla f_0(x)^T(y - x) = 0$, which satisfies the optimality condition (2.29) with equality.

Modeling convex optimization problems

Based on the atomic functions with known convexity and functional operations that preserve convexity, as introduced in the previous sections, the convexity of a wide range of convex optimization problems can now be verified following a standard routine, named *constructive convex analysis*. Specifically, each objective and constraint function of a convex optimization problem can be represented as a *composition tree* consisting of atomic functions and functional operations. To verify the convexity

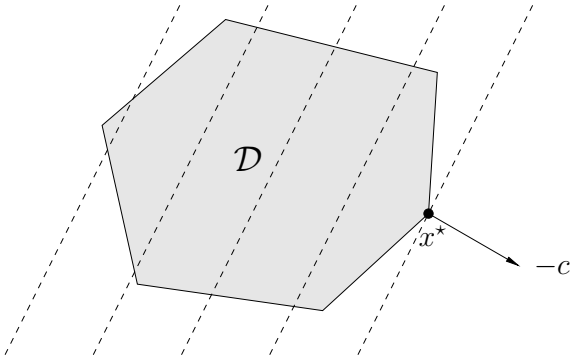

Figure 2.24 Geometric interpretation of an LP in $\mathbf{R}^2$. The goal is to find a point x in the feasible polyhedron $\mathcal{D}$ (shown shaded) that minimizes the linear objective function $c^T x$. The level curves of the objective function, which are hyperplanes orthogonal to c, are shown as dashed lines. The optimal point $x^\star$ corresponds to the point in $\mathcal{D}$ as far as possible in the direction of $-c$.

of a convex optimization problem, we can traverse the composition tree of each objective and constraint function from the leaves to the root, and at each node, check whether the convexity is preserved according to a series of rules. (See remark 4.4 for a simple example.)

These rules and routine form the *disciplined convex programming* framework for modeling convex optimization problems, so that any problem following these rules can be automatically verified (by a human or some computer software) as being convex. We refer the interested reader to appendix B for more details about these ideas.

2.5.3 Examples

Linear programs

When the objective and constraint functions of the convex optimization problem (2.26) are all affine, the problem is called a *linear program* (LP), which has the general form:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Gx \preceq h \\ & Ax = b \end{array}$$

with variable $x \in \mathbf{R}^n$, where $G \in \mathbf{R}^{m\times n}$, $h \in \mathbf{R}^m$, $A \in \mathbf{R}^{p\times n}$, $b \in \mathbf{R}^p$ and $c \in \mathbf{R}^n$ are the problem data. The feasible set of an LP is the intersection of finite halfspaces and hyperplanes, which is hence a polyhedron. Therefore, we can interpret an LP geometrically as the problem of minimizing a linear function over a polyhedron. This is illustrated in figure 2.24.

Example 2.24 *Least ℓ_1-norm fitting.* Suppose we are given the data $A \in \mathbf{R}^{m\times n}$ and $b \in \mathbf{R}^m$. We want to find a vector $x \in \mathbf{R}^n$ with the smallest ℓ_1-norm that fits the

linear system $Ax = b$, which corresponds to the problem

$$\begin{array}{ll} \text{minimize} & \|x\|_1 \\ \text{subject to} & Ax = b \end{array} \tag{2.30}$$

with variable $x \in \mathbf{R}^n$. Assuming the matrix A always has full rank, when $m \geq n$, the problem (2.30) is either infeasible or there exists only one feasible point; when $m = n$, the only feasible point is $x = A^{-1}b$. The problem (2.30) is interesting only when $m < n$, where the equation $Ax = b$ is underdetermined, *i.e.*, there are infinitely many feasible points.

We can transform the problem (2.30) into an equivalent LP by introducing an additional variable $t \in \mathbf{R}^n$ as

$$\begin{array}{ll} \text{minimize} & \mathbf{1}^T t \\ \text{subject to} & -t \preceq x \preceq t \\ & Ax = b, \end{array}$$

where the variables are now $x, t \in \mathbf{R}^n$. Another way to transform the problem (2.30) into an LP is to split x into its positive and negative parts, *i.e.*, let $x = x_+ - x_-$ with $x_+, x_- \succeq 0$, then we have

$$\begin{array}{ll} \text{minimize} & \mathbf{1}^T x_+ + \mathbf{1}^T x_- \\ \text{subject to} & x_+ \succeq 0, \quad x_- \succeq 0 \\ & A(x_+ - x_-) = b \end{array}$$

with variables $x_+, x_- \in \mathbf{R}^n$.

Quadratic programs

The convex optimization problem (2.26) is called a *quadratic program* (QP) if the objective function is a convex quadratic function, and the constraint functions are all affine, which has the general form:

$$\begin{array}{ll} \text{minimize} & (1/2)x^T P x + q^T x + r \\ \text{subject to} & Gx \preceq h \\ & Ax = b \end{array} \tag{2.31}$$

with variable $x \in \mathbf{R}^n$, where $P \in \mathbf{S}^n_+$, $q \in \mathbf{R}^n$, $r \in \mathbf{R}$, $G \in \mathbf{R}^{m \times n}$, $h \in \mathbf{R}^m$, $A \in \mathbf{R}^{p \times n}$ and $b \in \mathbf{R}^p$ are the problem data. The feasible set of a QP is also the intersection of finite halfspaces and hyperplanes, and hence we can interpret a QP geometrically as the problem of minimizing a convex quadratic function over a polyhedron. This is illustrated in figure 2.25.

If the inequality constraint functions in (2.31) are convex quadratic instead of affine, then the problem is called a *quadratically constrained quadratic program* (QCQP), which has the general form:

$$\begin{array}{ll} \text{minimize} & (1/2)x^T P_0 x + q_0^T x + r_0 \\ \text{subject to} & (1/2)x^T P_i x + q_i^T x + r_i \leq 0, \quad i = 1, \ldots, m \\ & Ax = b, \end{array} \tag{2.32}$$

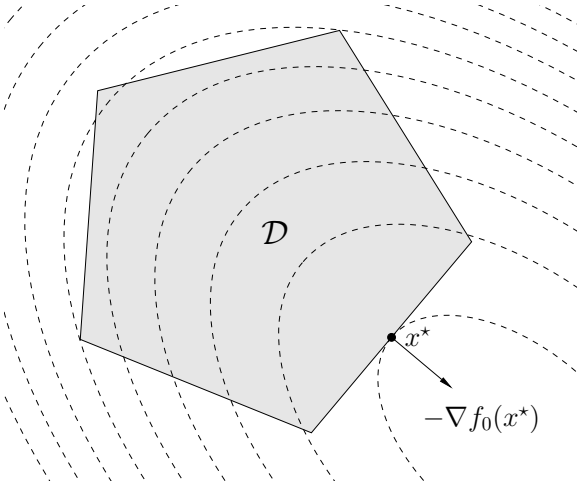

Figure 2.25 Geometric interpretation of a QP in $\mathbf{R}^2$. The goal is to find a point x in the feasible polyhedron $\mathcal{D}$ (shown shaded) that minimizes the convex quadratic objective function $f_0(x) = (1/2)x^T Px + q^T x + r$. The level curves of the objective function, which are (boundaries of) ellipsoids, are shown as dashed lines. The point $x^\star$ is optimal, and the arrow shows the direction of the negative gradient $-\nabla f_0(x^\star) = -(Px^\star + q)$.

where $P_0, P_1, \ldots, P_m \in \mathbf{S}^n_+$, $q_0, q_1, \ldots, q_m \in \mathbf{R}^n$, $r_0, r_1, \ldots, r_m \in \mathbf{R}$, $A \in \mathbf{R}^{p\times n}$ and $b \in \mathbf{R}^p$. In a QCQP, we minimize a convex quadratic function over a feasible region that is the intersection of ellipsoids (when $P_1, \ldots, P_m \succ 0$) and hyperplanes.

Note that QCQPs generalize QPs and LPs: When $P_1 = \cdots = P_m = 0$, the QCQP reduces to a QP, and if additionally $P_0 = 0$, the QCQP reduces to an LP.

Example 2.25 *Constrained least squares.* Suppose we are given the data $A \in \mathbf{R}^{m\times n}$ and $b \in \mathbf{R}^m$. The *least squares problem* is defined as

$$\text{minimize} \quad f_0(x) = \|Ax - b\|_2^2$$

with variable $x \in \mathbf{R}^n$. Least squares problems are (unconstrained) QPs, since the objective function can be expressed as

$$f_0(x) = x^T A^T Ax - 2b^T Ax + b^T b,$$

which is a convex quadratic function.

Now suppose we are given some prior information that x should lie within the unit Euclidean ball, *i.e.*, $\|x\|_2 \leq 1$. Then, the corresponding constrained least squares problem is given by

$$\begin{array}{ll} \text{minimize} & \|Ax - b\|_2^2 \\ \text{subject to} & \|x\|_2 \leq 1, \end{array}$$

which is a QCQP. To see this, notice that this problem can be expressed as

$$\begin{array}{ll} \text{minimize} & x^T A^T Ax - 2b^T Ax + b^T b \\ \text{subject to} & x^T x - 1 \leq 0, \end{array}$$

which corresponds to the general form QCQP (2.32) with $P_0 = A^T A$, $q_0 = -2A^T b$, $r_0 = b^T b$, $P_1 = I$, $q_1 = 0$, and $r_1 = -1$.

Semidefinite programs

An analogous of LPs in the space of symmetric matrices is called a *semidefinite program* (SDP), which has the general form:

$$\begin{array}{ll} \text{minimize} & \mathbf{tr}(CX) \\ \text{subject to} & \mathbf{tr}(A_i X) = b_i, \quad i = 1, \ldots, m \\ & X \succeq 0 \end{array} \tag{2.33}$$

with variable $X \in \mathbf{S}^n$, where $C, A_1, \ldots, A_m \in \mathbf{S}^n$ and $b \in \mathbf{R}^m$ are the problem data. Noticing that $\mathbf{tr}(AX)$ is the standard inner product of $\mathbf{S}^n$, which is a linear function of X for any fixed $A \in \mathbf{S}^n$ (see §A.3.1), the objective function and equality constraint functions of (2.33) are all affine. The feasible set of an SDP is then the intersection of an affine set and the positive semidefinite cone $\mathbf{S}^n_+$, which is a convex set. Hence, we can interpret an SDP geometrically as the problem of minimizing a linear function over the positive semidefinite cone intersected with an affine set.

An SDP is sometimes also expressed as:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & x_1 F_1 + \cdots + x_n F_n + G \preceq 0 \\ & Ax = b, \end{array} \tag{2.34}$$

where $x \in \mathbf{R}^n$ is the variable and $F_1, \ldots, F_n, G \in \mathbf{S}^n$, $A \in \mathbf{R}^{p \times n}$ and $b \in \mathbf{R}^p$ are the problem data. The inequality constraints are now linear matrix inequalities. It is easy to see from (2.34) that SDPs generalize LPs: When all matrices $F_1, \ldots, F_n, G$ are diagonal, the linear matrix inequality reduces to a set of n linear inequalities, and hence the SDP reduces to an LP.

Example 2.26 *Smallest eigenvalue of a symmetric matrix.* Recall that the smallest eigenvalue $\lambda_{\min}(A)$ of a symmetric matrix $A \in \mathbf{S}^n$ can be expressed as

$$\lambda_{\min}(A) = \inf_{z \neq 0} \frac{z^T A z}{z^T z}$$

(see §A.2.1). To find $\lambda_{\min}(A)$, we can solve the following SDP:

$$\begin{array}{ll} \text{maximize} & t \\ \text{subject to} & A - tI \succeq 0 \end{array} \tag{2.35}$$

with variables $t \in \mathbf{R}$. Let $t^\star$ be the optimal value (which is also the optimal point) of this problem, then we have $\lambda_{\min}(A) = t^\star$. To see this, notice that for any feasible point t of (2.35), we have $A - tI \succeq 0$, which implies that

$$z^T(A - tI)z \geq 0 \quad \Longleftrightarrow \quad z^T A z \geq t z^T z \quad \Longleftrightarrow \quad \frac{z^T A z}{z^T z} \geq t$$

for all $z \in \mathbf{R}^n$ and $z \neq 0$. Hence, we have $\lambda_{\min}(A) \geq t$ for all t that is feasible to (2.35), and in particular, $\lambda_{\min}(A) \geq t^\star$. On the other hand, for all $z \neq 0$, we have

$$\lambda_{\min}(A) \leq \frac{z^T A z}{z^T z} \quad \Longleftrightarrow \quad \lambda_{\min}(A) z^T z \leq z^T A z \quad \Longleftrightarrow \quad z^T (A - \lambda_{\min}(A) I) z \geq 0,$$

i.e., the matrix $A - \lambda_{\min}(A)I$ is positive semidefinite. Therefore, $\lambda_{\min}(A)$ is a feasible point of the SDP, and we have $t^\star \geq \lambda_{\min}(A)$. Combining the inequalities in both directions, we conclude that $\lambda_{\min}(A) = t^\star$.

Similarly, the optimal value of the problem

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & A - tI \preceq 0 \end{array}$$

with variable $t \in \mathbf{R}$ corresponds to the largest eigenvalue $\lambda_{\max}(A)$ of the symmetric matrix $A \in \mathbf{S}^n$.

Bibliographical notes

The concepts and properties about convex sets, convex functions, and convex optimization problems presented in this chapter are standard materials in *convex analysis*, which is an important subject of mathematics. For those claims and examples that we stated without proof, we refer the readers interested to the textbooks by Rockafellar [Roc70], Hiriart-Urruty and Lemaréchal [HL93a, HL93b, HL01], Bertsekas *et al.* [BNO03], Borwein and Lewis [BL06], as well as to the individual references listed in [BV04, chapter 1–5].

There are many excellent textbooks on various topics in mathematical optimization. Some general references include Polyak [Pol87], Nocedal and Wright [NW06], Luenberger and Ye [LY08], and Gill *et al.* [GMW19].

The standard reference for convex optimization is the textbook by Boyd and Vandenberghe [BV04], which covers a wide range of topics in convex optimization, including theory, algorithms, and applications. Other textbooks on convex optimization include Ben-Tal and Nemirovski [BN01], Nesterov [Nes04], Bertsekas [Ber09], and Kılınç-Karzan and Nemirovski [KN25]. The survey article by Vandenberghe and Boyd [VB96] provides a good introduction and extensive bibliographies to SDPs and their applications. See also the refrences listed in chapter 1.

The ideas of convex relaxation heuristics have been widely used since the early 2000s for finding an approximate solution to some nonconvex optimization problems. Two specific examples are the use of the nuclear norm as a convex surrogate for rank minimization and the ℓ_1-norm for cardinality minimization. Fazel [Faz02] showed that the nuclear norm is the convex envelope of the matrix rank function on the unit spectral norm ball, and as a special case of this, the vector ℓ_1-norm is the convex envelope of the cardinality function on the unit ℓ_∞-norm ball. Discussions and applications of these heuristics appear also in Fazel *et al.* [FHB03], Recht *et al.* [RFP10], and Chandrasekaran *et al.* [CRPW12].

There are several equivalent statements regarding the general composition rules for convex functions presented in §2.4.4, which could potentially be more useful in practice; see [BV04, §3.2.4] and [GBY06, §6.4].

The symbolic modeling of convex optimization problems has been a rather modern and popular research topic in recent years. The *disciplined convex programming* (DCP) framework was first introduced by Grant *et al.* [GBY06] as part of the CVX modeling system for MATLAB [GB14]. These ideas and software were further developed and extended in subsequent works, including Udell *et al.* [UMZ+14], Diamond and Boyd [DB16], Agrawal *et al.* [AVDB18], and Fu *et al.* [FNB20]. See also appendix B as well as the references therein.

Agrawal *et al.* [AVDB18] present a general framework for automatically transforming convex optimization problems expressed in a DCP-compliant way into standard forms that can be handled by generic numerical solvers, which has been implemented, *e.g.*, in the open-source software package CVXPY [DB16]. As a result, nowadays, the only requirement for a convex optimization practitioner in most application scenarios is to formulate their problem as a DCP-compliant convex program, while the rest of the steps (which are mostly tedious and error prone) can be automatically handled by a computer in the background.

We do not cover algorithms for solving convex optimization problems, which has been, over the years, more or less considered as a standard technique in many areas of applied mathematics and engineering. Interested readers may refer to Nesterov and Nemirovski [NN94], Bertsimas and Tsitsiklis [BT97], Boyd and Vandenberghe [BV04, part

III], Bertsekas [Ber15], Bubeck [Bub15], Nemirovski [Nem24], and the references listed in chapter 1 for more theoretical and technical details about how different classes of convex optimization problems can be handled efficiently in practice.

Numerical software packages for solving convex optimization problems are now widely available, and provide support for various classes of convex optimization problems, including LPs, QPs, QCQPs, SDPs, etc. Some popular open-source solvers for convex optimization problems include ECOS [DCB13, Dom13, Ser15], SCS [OCPB16, ZOB20, O'D21, OCPB23], GLPK [Mak20], OSQP [SBG+20, SNB+18, BGSB19, SBL20], and QOCO [CA25] in C, PROX-QP [BETC22] in C++, Clarabel [GC24, GCG20, CG23] in Rust, and CVXOPT [ADV04, Van10, ADLV11] in Python. There are also some commercial solvers, *e.g.*, Gurobi [Gur26], MOSEK [MOS26], and XPRESS [FIC26]. Just to name a few.

Exercises

Convex sets

2.1 *Expanded and restricted sets.* Let $S \subseteq \mathbf{R}^n$ be a set, and let $\|\cdot\|$ be a norm on $\mathbf{R}^n$.

(a) For any $\alpha \geq 0$, we define

$$S_\alpha = \left\{ x \;\middle|\; \inf_{y \in S} \|x - y\| \leq \alpha \right\}$$

as the *expansion* or *extension* of S by α. It is the set of all points whose distance to S is at most α. Show that if S is convex, then S_α is also convex.

(b) For any $\alpha \geq 0$, we define

$$S_{-\alpha} = \{x \mid B(x, \alpha) \subseteq S\},$$

where $B(x, \alpha)$ is the ball of radius α centered at x in the norm $\|\cdot\|$, as the *restriction* of S by α. It is the set of all points that are at least a distance of α away from $\mathbf{R}^n \setminus S$. Show that if S is convex, then $S_{-\alpha}$ is also convex.

2.2 *Separating hyperplane theorem.* Let $C, D \subseteq \mathbf{R}^n$ be two nonempty disjoint convex sets, *i.e.*, $C \cap D = \emptyset$.

(a) First assume that the (Euclidean) *distance* between C and D, given by

$$\mathbf{dist}(C, D) = \inf\{\|u - v\|_2 \mid u \in C, v \in D\},$$

is positive, and that there exists a pair of points $c \in C$ and $d \in D$ such that the distance between the sets is achieved, *i.e.*, $\|c - d\|_2 = \mathbf{dist}(C, D)$. Show that there exists a nonzero vector $a \in \mathbf{R}^n$ and a scalar $b \in \mathbf{R}$ such that

$$a^T x \leq b \text{ for all } x \in C$$

and

$$a^T x \geq b \text{ for all } x \in D.$$

In other words, the affine function $a^T x - b$ is nonpositive on C and nonnegative on D. (The hyperplane $\{x \mid a^T x = b\}$ is called a *separating hyperplane* of C and D, or is said to *separate* C and D.)

(b) Now show that that same conclusion in (a) also holds without the assumptions, *i.e.*, for any two nonempty disjoint convex sets C and D, there always exists a separating hyperplane of C and D. You may directly use the result proved in (a).
Hint.

i. If C and D are disjoint convex sets, then the set $\{x - y \mid x \in C, y \in D\}$ is also convex and does not contain the origin.

ii. You might also want to use the results from exercise 2.1.

2.3 *Strict separation.* If a separating hyperplane of two sets $C, D \subseteq \mathbf{R}^n$ satisfies the stronger consition that $a^T x < b$ for all $x \in C$ and $a^T x > b$ for all $x \in D$, then the hyperplane is said to *strictly separate* C and D.

(a) Let C be a closed convex set and $x_0 \notin C$. Show that there exists a hyperplane that strictly separates C and the point x_0.

(b) Give an example of two closed convex sets that are disjoint but cannot be strictly separated.

2.4 *Supporting hyperplane theorem.* Suppose $C \subseteq \mathbf{R}^n$ and x_0 is a point in the boundary of C. If $a \neq 0$ staisfies $a^T x \leq a^T x_0$ for all $x \in C$, then the hyperplane $\{x \mid a^T x = a^T x_0\}$ is called a *supporting hyperplane* of C at the point x_0. Show that for any convex set C and any point x_0 in the boundary of C, there always exists a supporting hyperplane of C at x_0. (This is known as the *supporting hyperplane theorem*.)
Hint. Use the separating hyperplane theorem from exercise 2.2.

2.5 *Convex sets as intersections of halfspaces.* Show that any closed convex set $C \subseteq \mathbf{R}^n$ can be expressed as the intersection of all halfspaces that contain it, *i.e.*,

$$C = \bigcap \{H \mid H \text{ halfspace}, S \subseteq H\}.$$

Hint. Use the results from exercise 2.3.

2.6 *Second-order cone.* Show that the set

$$\{(x, t) \in \mathbf{R}^{n+1} \mid \|x\|_2 \leq t\}$$

is a convex cone. What does the set look like in $\mathbf{R}^3$? (This set is called the *second-order cone*, or *quadratic cone*. It is also sometimes called the *Lorentz cone*.)

Algebra of convex sets

2.7 *Perspective of a set.* We define the *perspective function* $P\colon \mathbf{R}^{n+1} \to \mathbf{R}^n$ with domain $\mathbf{dom}\, P = \mathbf{R}^n \times \mathbf{R}_{++}$ as $P(z, t) = z/t$. Let $C \subseteq \mathbf{dom}\, P$ be a set, then the *perspective* of C is defined as the image of C under P, *i.e.*,

$$P(C) = \{P(x) \mid x \in C\} = \{z/t \mid (z, t) \in C\}.$$

(a) Suppose C is a line segment in $\mathbf{dom}\, P$, show that $P(C)$ is also a line segment.

(b) Suppose C is a convex set in $\mathbf{dom}\, P$, show that $P(C)$ is also convex.

(c) Show that if the set $C \subseteq \mathbf{R}^n$ is convex, then the inverse image of C under the perspective function P, given by

$$P^{-1}(C) = \{(x, t) \in \mathbf{R}^{n+1} \mid x/t \in C,\ t > 0\},$$

is also convex.

Convex functions

2.8 *Convex functions as affine envelopes.* Show that any convex function $f\colon \mathbf{R}^n \to \mathbf{R}$ can be expressed as the pointwise supremum of all affine functions that underestimate f, *i.e.*,

$$f(x) = \sup\{a^T x + b \mid a^T y + b \leq f(y) \text{ for all } y \in \mathbf{R}^n\}.$$

Hint. You might want to use the supporting hyperplane theorem from exercise 2.4.

2.9 *Gibbs' inequality.* Suppose that $p, q \in \mathbf{R}^n$ are two probability distributions over a finite set $\{1, \ldots, n\}$, *i.e.*, satisfy $\mathbf{1}^T p = 1$, $\mathbf{1}^T q = 1$, and $p \succeq 0$, $q \succeq 0$. Show that we have

$$\sum_{i=1}^{n} p_i \log\left(\frac{p_i}{q_i}\right) \geq 0,$$

with equality if and only if $p = q$.
Hint. Use Jensen's inequality and the strict concavity of the logarithm function.

2.10 *Affine envelope.* Show that the convex envelope of some function $f\colon \mathbf{R}^n \to \mathbf{R}$ can be expressed as the pointwise supremum of all affine functions that underestimate f, *i.e.*,

$$\mathbf{conv}\, f(x) = \sup\{a^T x + b \mid a^T y + b \leq f(y) \text{ for all } y \in \mathbf{R}^n\}.$$

Hint. Use the result from exercise 2.8.

2.11 *Log-concave functions.* A function $f\colon \mathbf{R}^n \to \mathbf{R}$ is said to be *log-concave* if $f(x) > 0$ for all $x \in \mathbf{dom}\, f$ and $\log f$ is a concave function. The *log-convex* functions are defined similarly.

(a) Show that a function $f\colon \mathbf{R}^n \to \mathbf{R}$ is log-convex if and only if $1/f$ is log-concave.

(b) Let $f\colon \mathbf{R}^n \to \mathbf{R}$ be a function with convex domain and $f(x) > 0$ for all $x \in \mathbf{dom}\, f$. Show that f is log-concave if and only if for all $x, y \in \mathbf{dom}\, f$ and $\theta \in [0, 1]$, we have

$$f(\theta x + (1-\theta)y) \geq f(x)^\theta f(y)^{1-\theta}.$$

In other words, the value of f at a convex combination of two points is at least the *geometric mean* of the values of f at those two points.

Functional operations

2.12 *Partial minimization.* Let $f\colon \mathbf{R}^n \times \mathbf{R}^m \to \mathbf{R}$ be a convex function (in particular, jointly convex in both variables), and $C \subseteq \mathbf{R}^m$ is a convex nonempty set. Show that the function $g\colon \mathbf{R}^n \to \mathbf{R}$ defined as

$$g(x) = \inf_{y \in C} f(x, y)$$

is a convex function, provided $g(x) > -\infty$ for all $x \in \mathbf{R}^n$.

2.13 *Perspective of a function.* Let $f\colon \mathbf{R}^n \to \mathbf{R}$ be a function, then the *perspective* of f is the function $g\colon \mathbf{R}^{n+1} \to \mathbf{R}$ defined as

$$g(x, t) = tf\,(x/t)$$

with domain

$$\mathbf{dom}\, g = \{(x, t) \mid x/t \in \mathbf{dom}\, f,\ t > 0\}.$$

(a) Suppose f is a convex function, show that its perspective g is also convex. Specifically, show that $\mathbf{dom}\, g$ is a convex set and g satisfies the Jensen's inequality (2.11).

(b) Show the results in (a) using epigraphs.

Hint. You may want to use the results from exercise 2.7.

2.14 *Relative entropy.* Show that the *relative entropy* function $f(x, y) = \sum_{i=1}^n x_i \log(x_i/y_i)$ is convex on $\mathbf{R}^n_{++} \times \mathbf{R}^n_{++}$.
Hint. Use the property of the perspective of a function from exercise 2.13.

2.15 *Infimal convolution.* Let $f, g\colon \mathbf{R}^n \to \mathbf{R}$ be two functions, then the *infimal convolution* of f and g is the function $h\colon \mathbf{R}^n \to \mathbf{R}$ defined as

$$h(x) = \inf\{f(y) + g(x - y) \mid y \in \mathbf{R}^n\}, \tag{2.36}$$

and is sometimes denoted as $h = f \,\square\, g$. We assume that $h(x) > -\infty$ and the infimum is achieved for all $x \in \mathbf{R}^n$.

(a) Show that if f and g are convex functions, then their infimal convolution $h = f \,\square\, g$ given by (2.36) is also a convex function.

(b) Show that the epigraph of $h = f \,\square\, g$ given by (2.36) is the *Minkowski sum* (see §A.1) of the epigraphs of f and g, *i.e.*,

$$\begin{aligned} \mathbf{epi}(f \,\square\, g) &= \mathbf{epi}\, f + \mathbf{epi}\, g \\ &= \{(x, t) = (y, s) + (z, r) \mid (y, s) \in \mathbf{epi}\, f,\ (z, r) \in \mathbf{epi}\, g\}. \end{aligned}$$

Convex optimization problems

2.16 *Second-order cone program.* A second-order cone program (SOCP) is defined as

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & \|A_i x + b_i\|_2 \leq c_i^T x + d_i, \quad i = 1, \ldots, m \\ & Fx = g, \end{array}$$

where $x \in \mathbf{R}^n$ is the variable, $A_i \in \mathbf{R}^{p_i \times n}$, $b_i \in \mathbf{R}^{p_i}$, $c_i \in \mathbf{R}^n$, $d_i \in \mathbf{R}$, $F \in \mathbf{R}^{q \times n}$ and $g \in \mathbf{R}^q$ are the problem data.

(a) Show that an SOCP is a convex optimization problem.

(b) When does an SOCP reduce to an LP or QCQP?

(We will see in the later chapters that, in fact, many convex optimization problems appear in applications can be formulated as SOCPs.)

Chapter 3

Sequential convex programming

3.1 Problems involving biconvex functions

We start from dealing with a relatively simple case of nonconvex optimization problems, where the variables can be divided into two blocks, and the objective function and constraints are convex in each block of variables when the other block is fixed.

3.1.1 Biconvex sets and functions

Biconvex sets

A set $B \subseteq \mathbf{R}^n \times \mathbf{R}^k$ is called a *biconvex set*, if for every fixed $\tilde{y} \in \mathbf{R}^k$, the set

$$B_{\tilde{y}} = \{x \in \mathbf{R}^n \mid (x, \tilde{y}) \in B\} \subseteq \mathbf{R}^n$$

is convex, and for every fixed $\tilde{x} \in \mathbf{R}^n$, the set

$$B_{\tilde{x}} = \{y \in \mathbf{R}^k \mid (\tilde{x}, y) \in B\} \subseteq \mathbf{R}^k$$

is convex.

Obviously, a biconvex set is not necessarily convex in general. In particular, a biconvex set does not even have to be connected, as shown by the example

$$B = \{(x, y) \in \mathbf{R}^2 \mid x, y < 0\} \cup \{(x, y) \in \mathbf{R}^2 \mid x, y > 0\},$$

which is the union of the positive and negative orthants in $\mathbf{R}^2$. Examples of some biconvex sets are shown in figure 3.1, and the set B defined above is shown on the right.

Example 3.1 *Solution set of bilinear equations.* Let $A \in \mathbf{R}^{m \times n}$, $C \in \mathbf{R}^{m \times k}$, and $b, d \in \mathbf{R}^m$. The solution set of a system of *bilinear equations*, given by

$$B = \{(x, y) \in \mathbf{R}^n \times \mathbf{R}^k \mid (Ax + b)^T (Cy + d) = 0\},$$

is a biconvex set. To see this, for every fixed $\tilde{y} \in \mathbf{R}^k$, we have

$$B_{\tilde{y}} = \{x \in \mathbf{R}^n \mid (Ax + b)^T (C\tilde{y} + d) = 0\}$$

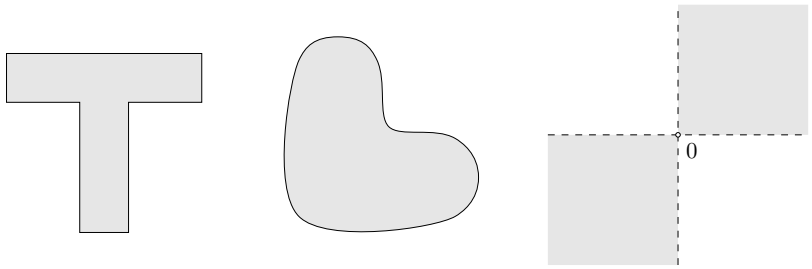

Figure 3.1 Examples of biconvex sets in $\mathbf{R}^2$. The union of the positive and negative orthants (*right*) is a biconvex set that is not connected.

$$= \{x \in \mathbf{R}^n \mid (C\tilde{y} + d)^T Ax = -(C\tilde{y} + d)^T b\},$$

which is an affine set in $\mathbf{R}^n$, and hence convex. Similarly, for every fixed $\tilde{x} \in \mathbf{R}^n$, we have

$$\begin{aligned} B_{\tilde{x}} &= \{y \in \mathbf{R}^k \mid (A\tilde{x} + b)^T (Cy + d) = 0\} \\ &= \{y \in \mathbf{R}^k \mid (A\tilde{x} + b)^T Cy = -(A\tilde{x} + b)^T d\}, \end{aligned}$$

which is also an affine set in $\mathbf{R}^k$. According to the definition of biconvex sets, the set B is therefore biconvex.

One of the most important algebraic operations that preserves biconvexity is that, similar to convex sets, the intersection of an arbitrary collection of biconvex sets is still biconvex.

Biconvex functions

A function $f : \mathbf{R}^n \times \mathbf{R}^k \to \mathbf{R}$ is a *biconvex function* if $\mathbf{dom}\, f$ is a biconvex set, and for every fixed $\tilde{y} \in \mathbf{R}^k$, the function

$$f_{\tilde{y}} : \mathbf{R}^n \to \mathbf{R}, \quad x \mapsto f(x, \tilde{y}),$$

is convex in x, and for every fixed $\tilde{x} \in \mathbf{R}^n$, the function

$$f_{\tilde{x}} : \mathbf{R}^k \to \mathbf{R}, \quad y \mapsto f(\tilde{x}, y),$$

is convex in y. In other words, a biconvex function is the one that is convex in each of two blocks of variables when the other block is fixed. We can also define *biconcave*, *biaffine*, and *bilinear* functions similarly, by replacing the property of being convex for $f_{\tilde{y}}$ and $f_{\tilde{x}}$ by the property of being concave, affine, or linear, respectively.

Example 3.2 *Biconvex functions.*

- The function $f : \mathbf{R}^n \times \mathbf{R}^n \to \mathbf{R}$ given by $f(x, y) = x^T y$ is bilinear (and hence biconvex), since for every fixed $\tilde{y} \in \mathbf{R}^n$, the function $f_{\tilde{y}}(x) = \tilde{y}^T x$ is linear in x, and for every fixed $\tilde{x} \in \mathbf{R}^n$, the function $f_{\tilde{x}}(y) = \tilde{x}^T y$ is linear in y. Figure 3.2 shows the graph of f when $n = 1$.

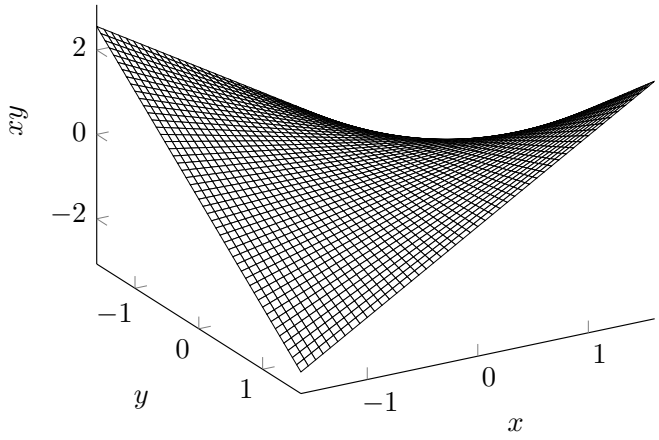

Figure 3.2 Graph of the bilinear function $f(x, y) = xy$.

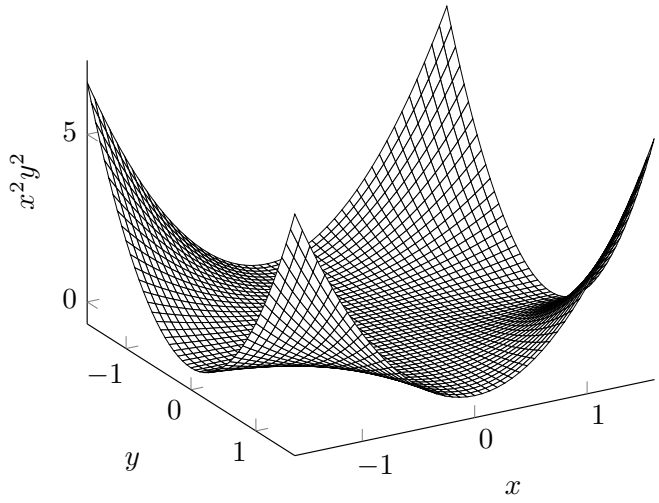

Figure 3.3 Graph of the biconvex function $f(x, y) = x^2y^2$.

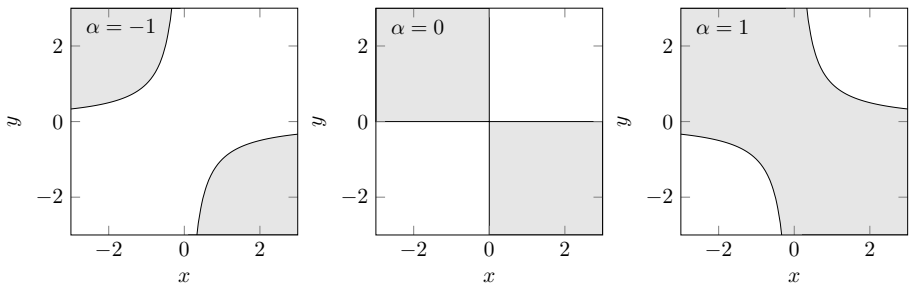

Figure 3.4 Sublevel sets of $f(x, y) = xy$ for different α (shown shaded).

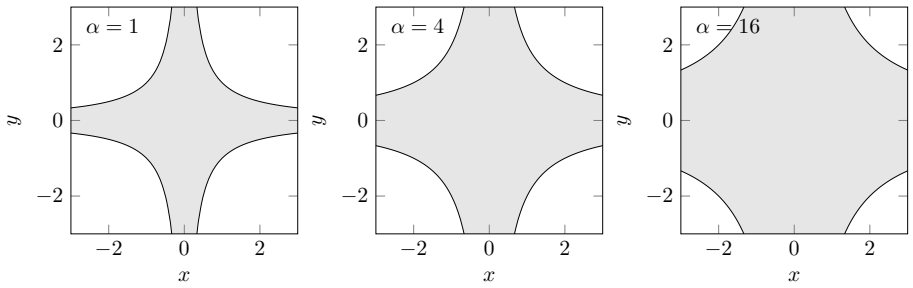

Figure 3.5 Sublevel sets of $f(x, y) = x^2y^2$ for different α (shown shaded).

- The function $f\colon \mathbf{R}^n \times \mathbf{R}^k \to \mathbf{R}$ given by $f(x, y) = (Ax + b)^T(Cy + d)$, where $A \in \mathbf{R}^{m\times n}$, $C \in \mathbf{R}^{m\times k}$, and $b, d \in \mathbf{R}^m$, is biaffine (and hence biconvex).
- The function $f\colon \mathbf{R}^{n\times m} \times \mathbf{R}^{m\times k} \to \mathbf{R}^{n\times k}$ given by $f(X, Y) = XY - A$ is biaffine.
- The function $f\colon \mathbf{R} \times \mathbf{R} \to \mathbf{R}$ given by $f(x, y) = x^2y^2$ is biconvex, since for every fixed $\tilde{y} \in \mathbf{R}$, the function $f_{\tilde{y}}(x) = \tilde{y}^2x^2$ is a convex quadratic in x, and for every fixed $\tilde{x} \in \mathbf{R}$, the function $f_{\tilde{x}}(y) = \tilde{x}^2y^2$ is a convex quadratic in y. The graph of f is shown in figure 3.3.

Similar to convex functions, the sublevel sets of a biconvex function $f\colon \mathbf{R}^n \times \mathbf{R}^k \to \mathbf{R}$, given by

$$C_\alpha = \{(x, y) \in \mathbf{dom}\, f \mid f(x, y) \leq \alpha\},$$

are biconvex sets for all $\alpha \in \mathbf{R}$. Figures 3.4 and 3.5 show the sublevel sets of the bilinear function $f(x, y) = xy$ and the biconvex function $f(x, y) = x^2y^2$, respectively, for different values of α.

Functional operations

Many operations that preserves convexity can be transferred to biconvex functions:

- *Nonnegative weighted sums.* Evidently, if f is a biconvex function and $w \geq 0$, then the function wf is biconvex. If f_1 and f_2 are both biconvex functions,

then so is their sum $f_1 + f_2$. In the general case, if $f_1, \ldots, f_m$ are biconvex functions, and $w_1, \ldots, w_m \geq 0$, then the function

$$f = w_1 f_1 + \cdots + w_m f_m$$

is also biconvex.

- *Pointwise maximum and supremum.* If $f_1, \ldots, f_m$ are biconvex functions, then the function

$$f(x, y) = \max\{f_1(x, y), \ldots, f_m(x, y)\}$$

is also biconvex. More generally, if $\{f_i\}_{i \in I}$ is a family of biconvex functions indexed by a set $\mathcal{I}$, then the function

$$f(x, y) = \sup_{i \in \mathcal{I}} f_i(x, y)$$

is biconvex.

- *Composition with a biaffine mapping.* Suppose $h\colon \mathbf{R}^m \to \mathbf{R}$ is convex, and $g\colon \mathbf{R}^n \times \mathbf{R}^k \to \mathbf{R}^m$ is biaffine, then the function $f\colon \mathbf{R}^n \times \mathbf{R}^k \to \mathbf{R}$, given by $f(x, y) = h(g(x, y))$ is biconvex. In particular, if $h\colon \mathbf{R} \to \mathbf{R}$ is convex, and $A \in \mathbf{R}^{m \times n}$, $C \in \mathbf{R}^{m \times k}$, and $b, d \in \mathbf{R}^m$, then the function $f\colon \mathbf{R}^n \times \mathbf{R}^k \to \mathbf{R}$, given by

$$f(x, y) = h((Ax + b)^T (Cy + d))$$

is biconvex.

- *General composition.* If $h\colon \mathbf{R} \to \mathbf{R}$ is convex and nondecreasing, and $g\colon \mathbf{R}^n \times \mathbf{R}^k \to \mathbf{R}$ is biconvex, then the function $f\colon \mathbf{R}^n \times \mathbf{R}^k \to \mathbf{R}$, given by $f(x, y) = h(g(x, y))$ is biconvex. This property can be easily extended to multivariate functions $h\colon \mathbf{R}^m \to \mathbf{R}$ that are, *e.g.*, convex and nondecreasing in each argument (see §2.4.4).

Example 3.3 *Matrix factorization cost.* Consider the function $f\colon \mathbf{R}^{n \times m} \times \mathbf{R}^{m \times k} \to \mathbf{R}$ given by

$$f(X, Y) = \|XY - A\|_F^2 = \sum_{i=1}^{n} \sum_{j=1}^{k} ((XY)_{ij} - A_{ij})^2,$$

where $A \in \mathbf{R}^{n \times k}$ is some given data. We can express f as the composition of the convex function $h\colon \mathbf{R}^{n \times k} \to \mathbf{R}$, given by $h(Z) = \|Z\|_F^2 = \sum_{i=1}^{n} \sum_{j=1}^{k} Z_{ij}^2$, with the biaffine function $g\colon \mathbf{R}^{n \times m} \times \mathbf{R}^{m \times k} \to \mathbf{R}^{n \times k}$, given by $g(X, Y) = XY - A$. Hence, by the composition property above, f is biconvex.

3.1.2 Biconvex optimization problems

We call an optimization problem that involves biconvex functions in the following general form a *biconvex optimization problem* (or *biconvex program*):

$$\begin{array}{ll} \text{minimize} & f_0(x, y) \\ \text{subject to} & f_i(x, y) \leq 0, \quad i = 1, \ldots, m \\ & h_i(x, y) = 0, \quad i = 1, \ldots, p, \end{array} \tag{3.1}$$

where $x \in \mathbf{R}^n$, $y \in \mathbf{R}^k$ are the optimization variables. The functions $f_i \colon \mathbf{R}^n \times \mathbf{R}^k \to \mathbf{R}$, $i = 0, \ldots, m$, are biconvex, and $h_i \colon \mathbf{R}^n \times \mathbf{R}^k \to \mathbf{R}$, $i = 1, \ldots, p$, are biaffine. The feasible set of (3.1) is given by

$$\mathcal{D} = \left\{ (x, y) \in \bigcap_{i=0}^{m} \mathbf{dom}\, f_i \cap \bigcap_{i=1}^{p} \mathbf{dom}\, h_i \;\middle|\; \begin{array}{ll} f_i(x, y) \leq 0, & i = 1, \ldots, m \\ h_i(x, y) = 0, & i = 1, \ldots, p \end{array} \right\},$$

which is a biconvex set. To see this, notice that the domain of the objective and of each constraint function is biconvex, each of the constraints defines a biconvex set, and finally the intersection of biconvex sets remains biconvex. Hence, similar to convex optimization problems, the biconvex problem (3.1) can be interpreted as minimizing some biconvex objective function f_0 over a biconvex set.

Optimal value and points

Different from convex optimization problems, in the most general case, very little can be said about the global or even local optimality properties of biconvex optimization problems. Instead, the notion of *partial optimality*, which is even weaker than local optimality, is the most commonly considered criterion of optimality for biconvex optimization problems. Suppose $(x^\star, y^\star) \in \mathcal{D}$ is a feasible point of (3.1), then $(x^\star, y^\star)$ is a *partially optimal point* of (3.1) if for all $x \in \mathcal{D}_{y^\star}$, $y \in \mathcal{D}_{x^\star}$, we have

$$f_0(x^\star, y^\star) \leq f_0(x, y^\star) \qquad \text{and} \qquad f_0(x^\star, y^\star) \leq f_0(x^\star, y),$$

where

$$\mathcal{D}_{y^\star} = \left\{ x \in \mathbf{R}^n \;\middle|\; \begin{array}{ll} (x, y^\star) \in \mathbf{dom}\, f_0 & \\ f_i(x, y^\star) \leq 0, & i = 1, \ldots, m \\ h_i(x, y^\star) = 0, & i = 1, \ldots, p \end{array} \right\},$$

and

$$\mathcal{D}_{x^\star} = \left\{ y \in \mathbf{R}^k \;\middle|\; \begin{array}{ll} (x^\star, y) \in \mathbf{dom}\, f_0 & \\ f_i(x^\star, y) \leq 0, & i = 1, \ldots, m \\ h_i(x^\star, y) = 0, & i = 1, \ldots, p \end{array} \right\}.$$

The objective value $f_0(x^\star, y^\star)$ is called the *partial optimal value* at the point $(x^\star, y^\star)$.

It can be shown that for a differentiable biconvex optimization problem, every stationary point of f_0 over $\mathcal{D}$ is partially optimal, and vice versa. However, such a point is not necessarily a local optimum, as stationary points can be saddle points of the objective function. Despite a weak notion of optimality, partially optimal points for biconvex problems is still widely used in practice, and turns out to be good enough for many applications.

3.1.3 Alternate convex search

As a result of the optimality properties regarding biconvex optimization problems discussed in the last section, 'solving' a biconvex optimization problem in practice usually resolves to finding a stationary point of the objective function over the

feasible set. In this section, we introduce a heuristic for this purpose based on *alternate convex search* (ACS) that is widely considered in practice.

The basic idea of ACS is to transform the biconvex problem into two convex subproblems, which can then be handled directly by solution methods for convex optimization. Specifically, for a biconvex optimization problem of the form (3.1), ACS iterates between solving the following two convex subproblems:

$$\begin{array}{ll} \text{minimize} & f_0(x,\tilde{y}) \\ \text{subject to} & f_i(x,\tilde{y}) \leq 0, \quad i=1,\ldots,m \\ & h_i(x,\tilde{y}) = 0, \quad i=1,\ldots,p, \end{array} \tag{3.2}$$

where $x \in \mathbf{R}^n$ is the variable, $\tilde{y} \in \mathbf{R}^k$ is the fixed problem data, and

$$\begin{array}{ll} \text{minimize} & f_0(\tilde{x},y) \\ \text{subject to} & f_i(\tilde{x},y) \leq 0, \quad i=1,\ldots,m \\ & h_i(\tilde{x},y) = 0, \quad i=1,\ldots,p, \end{array} \tag{3.3}$$

where $y \in \mathbf{R}^k$ is the variable, $\tilde{x} \in \mathbf{R}^n$ is the fixed data. Since the problems (3.2) and (3.3) are convex, efficient convex minimization methods can be used to solve these subproblems. The full ACS procedure is summarized in the following algorithm.

Algorithm 3.1 ALTERNATE CONVEX SEARCH.

given a starting point $(x^{(0)}, y^{(0)})$ feasible to (3.1).

$k := 0$.

repeat

1. *Solve (3.2) with* $\tilde{y} = y^{(k)}$.

$$x^{(k+1)} := \operatorname*{argmin}_{x \in \mathbf{R}^n} \left\{ f_0(x, y^{(k)}) \;\middle|\; \begin{array}{l} f_i(x, y^{(k)}) \leq 0, \; i=1,\ldots,m \\ h_i(x, y^{(k)}) = 0, \; i=1,\ldots,p \end{array} \right\}. \tag{3.4}$$

2. *Solve (3.3) with* $\tilde{x} = x^{(k+1)}$.

$$y^{(k+1)} := \operatorname*{argmin}_{y \in \mathbf{R}^k} \left\{ f_0(x^{(k+1)}, y) \;\middle|\; \begin{array}{l} f_i(x^{(k+1)}, y) \leq 0, \; i=1,\ldots,m \\ h_i(x^{(k+1)}, y) = 0, \; i=1,\ldots,p \end{array} \right\}. \tag{3.5}$$

3. $k := k+1$.

until stopping criteria is satisfied.

The order of solving the two subproblems (3.2) and (3.3) in each iteration of ACS can be swapped, *i.e.*, in algorithm 3.1, one may first perform the update of y according to (3.5), and then update x according to (3.4).

Remark 3.1 *Stopping criteria.* Let $\epsilon > 0$ be some small threshold. There are several ways to define a stopping criteria for the ACS iterations.

One choice is the difference of the objective values of (3.1) with the variable values obtained between two consecutive iterations is below a certain threshold $\epsilon > 0$, *i.e.*, quit when

$$|f_0(x^{(k+1)}, y^{(k+1)}) - f_0(x^{(k)}, y^{(k)})| < \epsilon. \tag{3.6}$$

As a small variation to (3.6), one may also use the difference between the optimal values of the problems (3.2) and (3.3) in one iteration as the criterion, *i.e.*, quit when

$$|f_0(x^{(k+1)}, y^{(k+1)}) - f_0(x^{(k+1)}, y^{(k)})| < \epsilon. \tag{3.7}$$

In practice, there is not much difference between using (3.6) and (3.7) as the termination criteria of the ACS procedure, except that using the latter does not require storing the objective value of (3.1) from the previous iteration.

Other choices include limiting the maximum number of iterations, or stopping when the changes of the optimization variables are below certain thresholds, *e.g.*, quit when

$$\max\{\|x^{(k+1)} - x^{(k)}\|_2,\ \|y^{(k+1)} - y^{(k)}\|_2\} < \epsilon.$$

The stopping criterion may also depend on the special structure of the given biconvex objective function.

It is not hard to see that (under some technical conditions), if $(x^{(k)}, y^{(k)})$ is feasible to (3.1), then $(x^{(k+1)}, y^{(k+1)})$ after one ACS iteration is also feasible, and the sequence of objective function values $\{f_0(x^{(k)}, y^{(k)})\}_{k=0}^{\infty}$ is monotonically nonincreasing and hence convergent. Furthermore, if the sequence $\{(x^{(k)}, y^{(k)})\}_{k=0}^{\infty}$ converges to $(x^\star, y^\star)$, then $(x^\star, y^\star)$ is a stationary point of f_0 (and hence, is a partially optimal point of (3.1)).

Remark 3.2 *Initialization.* Algorithm 3.1 as a heuristic method is sensitive to initialization, *i.e.*, with different initial points $(x^{(0)}, y^{(0)})$ that is feasible, the sequence $\{(x^{(k)}, y^{(k)})\}_{k=0}^{\infty}$ generated through the ACS procedure may converge to different stationary points of f_0 with (possibly) different function values.

Example 3.4 *Nonnegative matrix factorization.* We illustrate the ACS procedure through a simple biconvex optimization example. Consider the following *nonnegative matrix factorization* problem:

$$\begin{array}{ll} \text{minimize} & \|XY - A\|_F^2 \\ \text{subject to} & X_{ij} \geq 0, \quad i = 1, \ldots, m, \quad j = 1, \ldots, k \\ & Y_{ij} \geq 0, \quad i = 1, \ldots, k, \quad j = 1, \ldots, n \end{array} \tag{3.8}$$

with variables $X \in \mathbf{R}^{m \times k}$ and $Y \in \mathbf{R}^{k \times n}$, where $A \in \mathbf{R}^{m \times n}$ is some given data matrix. This problem is a biconvex optimization problem, since the objective function $\|XY - A\|_F^2$ is biconvex, and the constraints are all affine (and of course, biaffine).

We take $m = n = 50$ and $k = 5$, and the data matrix A is generated randomly. Figure 3.6 shows the convergence behavior of the ACS procedure (algorithm 3.1) applied to the problem (3.8) with 5 different random initializations. We see that the objective values is always nonincreasing over iterations, and almost all runs converge within 25 iterations. It is also obvious in this example that the ACS procedure is initialization sensitive, *i.e.*, starting from different initial values of the variables may lead to different final convergent points.

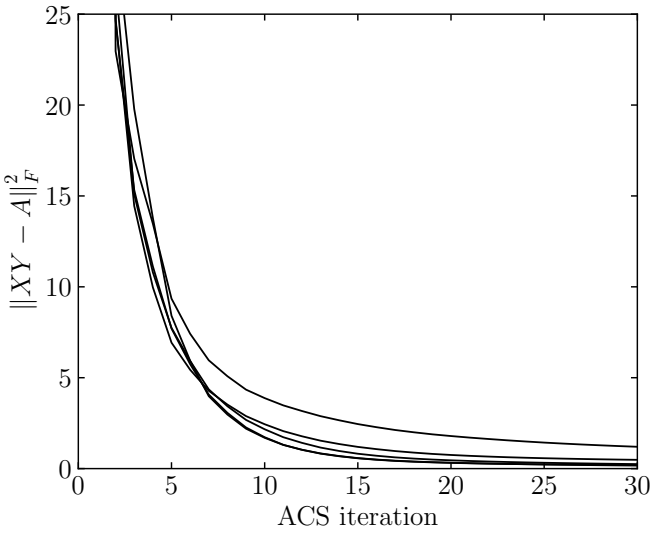

Figure 3.6 Convergence behavior of the ACS procedure applied to the nonnegative matrix factorization problem (3.8) with 5 different random initializations.

3.2 Difference-of-convex programming

Another important class of nonconvex optimization problems which can be handled by convex programming techniques involve objective and constraint functions that can be expressed as the difference of two convex functions.

3.2.1 Difference-of-convex functions

A *difference-of-convex function* $h\colon \mathbf{R}^n \to \mathbf{R}$ is a function that can be expressed as the difference of two convex functions, *i.e.*, has the form

$$h(x) = f(x) - g(x), \tag{3.9}$$

where $f\colon \mathbf{R}^n \to \mathbf{R}$ and $g\colon \mathbf{R}^n \to \mathbf{R}$ are both convex functions. Note that any convex function f is also a difference-of-convex function, since it can be expressed as $f(x) = f(x) - 0$. Similarly, any concave function h is also a difference-of-convex function, since it can be expressed as $h(x) = 0 - (-h(x))$.

Example 3.5 *Difference-of-convex functions.* A simple example of difference-of-convex functions on $\mathbf{R}$ is given by $h(x) = x^4 + (1/5)x - x^2$, which can be expressed as the difference between the two convex functions $f(x) = x^4 + (1/5)x$ and $g(x) = x^2$. The graph of this function is shown in figure 3.7.

An example of difference-of-convex functions on $\mathbf{R}^n$ is given by the following (possibly nonconvex) quadratic function:

$$h(x) = x^T P x + q^T x + r,$$

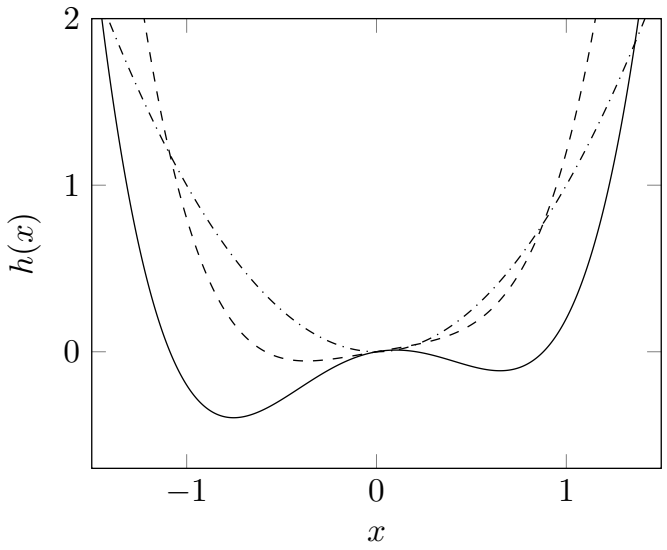

Figure 3.7 Graph of the difference-of-convex function $h(x) = x^4 + (1/5)x - x^2$ (shown solid), along with its convex components $f(x) = x^4 + (1/5)x$ (shown dashed) and $g(x) = x^2$ (shown dashdotted).

where $P \in \mathbf{S}^n$, $q \in \mathbf{R}^n$, and $r \in \mathbf{R}$. Let $P = Q\Lambda Q^T$ be the eigenvalue decomposition of P (see §A.2.1), where $Q \in \mathbf{R}^{n\times n}$ is orthogonal, and $\Lambda = \mathbf{diag}(\lambda_1, \ldots, \lambda_n)$ is a diagonal matrix whose diagonal elements are the eigenvalues of P. We can decompose Λ as $\Lambda = \Lambda_+ - \Lambda_-$, where

$$\Lambda_+ = \mathbf{diag}(\max\{\lambda_1, 0\}, \ldots, \max\{\lambda_n, 0\})$$

and

$$\Lambda_- = \mathbf{diag}(\max\{-\lambda_1, 0\}, \ldots, \max\{-\lambda_n, 0\}).$$

Let $P_+ = Q\Lambda_+ Q^T$ and $P_- = Q\Lambda_- Q^T$, then we have $P_+, P_- \in \mathbf{S}^n_+$ and $P_+ - P_- = P$. Hence, we can express h as

$$h(x) = x^T P_+ x + q^T x + r - x^T P_- x,$$

which is the difference of two convex quadratic functions $f(x) = x^T P_+ x + q^T x + r$ and $g(x) = x^T P_- x$.

First-order majorization

Consider a difference-of-convex function $h\colon \mathbf{R}^n \to \mathbf{R}$ in the form (3.9). If the convex function g is differentiable at some point $\tilde{x} \in \mathbf{R}^n$, then by the first-order condition of convexity (2.13), we have

$$\hat{g}_{\tilde{x}}(x) = g(\tilde{x}) + \nabla g(\tilde{x})^T (x - \tilde{x}) \leq g(x)$$

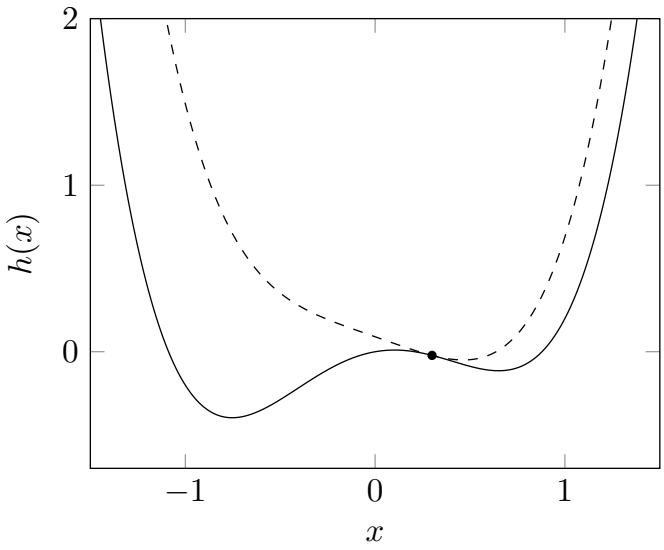

Figure 3.8 Graph of the first-order majorizer $\hat{h}_{\tilde{x}}(x) = x^4 + (1/5)x - 2\tilde{x}x + \tilde{x}^2$ (shown dashed) at the point $\tilde{x} = 0.3$ (shown circle) of the difference-of-convex function $h(x) = x^4 + (1/5)x - x^2$ (shown solid).

for all $x \in \mathbf{R}^n$, where $\hat{g}_{\tilde{x}} \colon \mathbf{R}^n \to \mathbf{R}$ is the first-order Taylor approximation of g at the point $\tilde{x}$. Define the function $\hat{h}_{\tilde{x}} \colon \mathbf{R}^n \to \mathbf{R}$ as

$$\hat{h}_{\tilde{x}}(x) = f(x) - \hat{g}_{\tilde{x}}(x) = f(x) - g(\tilde{x}) - \nabla g(\tilde{x})^T(x - \tilde{x}), \tag{3.10}$$

then we have

$$\hat{h}_{\tilde{x}}(x) \geq f(x) - g(x) = h(x) \tag{3.11}$$

for all $x \in \mathbf{R}^n$, *i.e.*, $\hat{h}_{\tilde{x}}$ is a global upper bound (or *majorizer*) of h. The first-order majorizer $\hat{h}_{\tilde{x}}$ is a convex function, since it is the difference between the convex function f and the affine function $\hat{g}_{\tilde{x}}$, and furthermore, notice that at the point $\tilde{x}$, we have

$$\hat{h}_{\tilde{x}}(\tilde{x}) = f(\tilde{x}) - g(\tilde{x}) = h(\tilde{x}),$$

which shows that such a majorization is *tight* at the point $\tilde{x}$.

As an example, consider the difference-of-convex function $h(x) = x^4 + (1/5)x - x^2$ on $\mathbf{R}$ from example 3.5, where the convex components are given by $f(x) = x^4 + (1/5)x$ and $g(x) = x^2$. The first-order majorization of h in the form (3.10) at the point $\tilde{x}$ is given by

$$\hat{h}_{\tilde{x}}(x) = x^4 + (1/5)x - (\tilde{x}^2 + 2\tilde{x}(x - \tilde{x})) = x^4 + (1/5)x - 2\tilde{x}x + \tilde{x}^2.$$

The graph of this first-order majorizer $\hat{h}_{\tilde{x}}$ at the point $\tilde{x} = 0.3$ is shown in figure 3.8.

3.2.2 Difference-of-convex optimization problems

A *difference-of-convex programming* problem has the following general form:

$$\begin{array}{ll} \text{minimize} & f_0(x) - g_0(x) \\ \text{subject to} & f_i(x) - g_i(x) \leq 0, \quad i = 1, \ldots, m, \end{array} \tag{3.12}$$

where $x \in \mathbf{R}^n$ is the optimization variable, and the functions $f_i \colon \mathbf{R}^n \to \mathbf{R}$ and $g_i \colon \mathbf{R}^n \to \mathbf{R}$ for $i = 0, \ldots, m$ are all convex.

Note that the standard form difference-of-convex problem (3.12) only involves inequality constrains, since equality constrains of the form

$$p_i(x) = q_i(x), \quad i = 1, \ldots, k,$$

where $p_i \colon \mathbf{R}^n \to \mathbf{R}$ and $q_i \colon \mathbf{R}^n \to \mathbf{R}$ are convex functions can be equivalently expressed in the difference-of-convex form in (3.12) as two inequality constraints

$$p_i(x) - q_i(x) \leq 0 \quad \text{and} \quad q_i(x) - p_i(x) \leq 0$$

for all $i = 1, \ldots, k$.

When g_i are all affine functions, the problem (3.12) reduces to a convex optimization problem and hence can be efficiently solved. In the most general case, however, difference-of-convex programming problems can be very hard to solve.

Example 3.6 *Boolean least squares.* Least squares problem with boolean constrains has the form

$$\begin{array}{ll} \text{minimize} & \|Ax - b\|_2^2 \\ \text{subject to} & x_i \in \{0, 1\}, \quad i = 1, \ldots, n, \end{array} \tag{3.13}$$

where $x \in \mathbf{R}^n$ is the optimization variable, $A \in \mathbf{R}^{m \times n}$ and $b \in \mathbf{R}^m$ are given data. To transform the problem (3.13) into the standard form (3.12), we notice that the boolean constraints is equivalent to the quadratic equations

$$x_i(1 - x_i) = 0, \quad i = 1, \ldots, n,$$

which can be further expressed as two inequality constraints as

$$x_i(1 - x_i) \leq 0 \quad \text{and} \quad -x_i(1 - x_i) \leq 0, \quad i = 1, \ldots, n.$$

Hence, the boolean least squares problem (3.13) in the standard difference-of-convex programming form is given by

$$\begin{array}{ll} \text{minimize} & \|Ax - b\|_2^2 \\ \text{subject to} & x_i^2 - x_i \leq 0, \quad i = 1, \ldots, n \\ & x_i - x_i^2 \leq 0, \quad i = 1, \ldots, n, \end{array}$$

with variable $x \in \mathbf{R}^n$. The first set of inequality constrains are the difference between the convex quadratic x_i^2 and the affine function x_i and are hence convex, while the second set of inequality constraints are concave in the difference-of-convex form.

Example 3.7 *Gaussian covariance estimation.* Consider a random vector $y \in \mathbf{R}^n$ from a multivariate Gaussian distribution with mean zero and covariance matrix $X = \mathbf{E}\, yy^T$, and $X \in \mathbf{S}^n_{++}$ is positive definite. Suppose we are given a dataset $y_i \in \mathbf{R}^n$, $i = 1, \ldots, m$, which are m independent samples drawn from this distribution, and our goal is to estimate the covariance matrix X from this dataset. This problem can be formulated as

$$\text{minimize} \quad \log\det X + \mathbf{tr}(X^{-1}Y), \tag{3.14}$$

with variable X (and implicit constraint $X \succ 0$), where $Y = (1/m)\sum_{i=1}^m y_i y_i^T$ is the sample covariance matrix (see §4.2.4). This problem is not convex (although a simple change of variable $S = X^{-1}$ transforms it into a convex problem), since the first term $\log\det X$ is concave. However, noticing that the second term $\mathbf{tr}(X^{-1}Y)$ is convex (with fixed Y), so the problem (3.14) can be trivially transformed into the difference-of-convex form as

$$\text{minimize} \quad \mathbf{tr}(X^{-1}Y) - (-\log\det X),$$

with variable X.

3.2.3 Convex-concave procedure

Similar to biconvex optimization problems, 'solving' a difference-of-convex programming problem in practice usually means finding a stationary point of the objective function over the feasible set.

A widely used heuristic for this purpose is the *convex-concave procedure* (CCP). The basic idea of CCP is to iteratively form the first-order majorization (3.10) to the objective and inequality constraint functions of the difference-of-convex problem (3.12) by linearizing g_i around the current point, and then solve the resulting convex optimization problem to update the variable values. Specifically, assume the problem (3.12) is differentiable, and let $\tilde{x} \in \mathbf{R}^n$ be the current value of the problem variable, then in the next CCP iteration, we solve the following approximation of the original problem:

$$\begin{array}{ll} \text{minimize} & f_0(x) - \hat{g}_{0,\tilde{x}}(x) \\ \text{subject to} & f_i(x) - \hat{g}_{i,\tilde{x}}(x) \leq 0, \quad i = 1, \ldots, m, \end{array} \tag{3.15}$$

where

$$\hat{g}_{i,\tilde{x}}(x) = g_i(\tilde{x}) + \nabla g_i(\tilde{x})^T(x - \tilde{x}) \tag{3.16}$$

for $i = 0, \ldots, m$ is the first-order approximation of g_i at the point $\tilde{x}$. The problem (3.15) is now a convex optimization problem, since the objective and constraint functions are all the difference between a convex function and an affine function.

Note that if the objective or some of the constraint functions in (3.12) are already convex, then we can always define the corresponding g_i to be the zero function, and the linearization (3.16) is also zero, so that these functions remain unchanged during all CCP iterations. In other words, convex components of the original difference-of-convex problem (3.12) are always preserved in the convex approximation (3.15) during CCP.

Interpretation

We can interpret the convex approximation (3.15) as follows: By solving the problem (3.15) at some $\tilde{x}$, we are finding a point that minimizes a global upper bound of the original objective function, over a convex restriction of the original feasible set.

The first part of this interpretation is clear according to the majorization inequality (3.11); to see the second part, notice that for any feasible $x \in \mathbf{R}^n$ of (3.15), *i.e.*, satisfying

$$f_i(x) - \hat{g}_{i,\tilde{x}}(x) \leq 0, \quad i = 1, \ldots, m,$$

we have

$$f_i(x) - g_i(x) \leq f_i(x) - \hat{g}_{i,\tilde{x}}(x) \leq 0$$

for all $i = 1, \ldots, m$. In other words, any feasible point of (3.15) must be feasible to the original problem (3.12).

Example 3.8 *Convex restrictions in CCP.* Consider the following (nonconvex) constraint:

$$\|x\|_2 \geq 1 \tag{3.17}$$

in the variable $x \in \mathbf{R}^n$. Geometrically, this constraint defines the set of points outside the unit ball centered at the origin. Figure 3.9 shows the constraint (3.17) in $\mathbf{R}^2$, where the corresponding feasible set is shown shaded.

We can rewrite the nonconvex inequality constraint (3.17) in the difference-of-convex form as

$$0 - (\|x\|_2 - 1) \leq 0, \tag{3.18}$$

where the two convex components are given by $f(x) = 0$ and $g(x) = \|x\|_2 - 1$. Now suppose we are at some point $\tilde{x} \in \mathbf{R}^n$ with $\|\tilde{x}\|_2 \geq 1$, and let

$$\hat{g}_{\tilde{x}}(x) = g(\tilde{x}) + \nabla g(\tilde{x})^T(x - \tilde{x}) = (\|\tilde{x}\|_2 - 1) + \frac{\tilde{x}^T}{\|\tilde{x}\|_2}(x - \tilde{x})$$

be the first-order approximation of g at the point $\tilde{x}$. Then the first-order majorization in the form (3.10) of the left-hand side of (3.18) at the point $\tilde{x}$ is given by

$$f(x) - \hat{g}_{\tilde{x}}(x) = 1 - \|\tilde{x}\|_2 - \frac{\tilde{x}^T}{\|\tilde{x}\|_2}(x - \tilde{x}) = 1 - \frac{\tilde{x}^T}{\|\tilde{x}\|_2}x,$$

so in the next CCP iteration, the convex restriction $f(x) - \hat{g}_{\tilde{x}}(x) \leq 0$ of the original nonconvex constraint (3.18) is given by

$$\frac{\tilde{x}^T}{\|\tilde{x}\|_2}x \geq 1. \tag{3.19}$$

Geometrically, $(1/\|\tilde{x}\|_2)\tilde{x}^T x = 1$ defines a tangent hyperplane of the unit ball at the point $\tilde{x}/\|\tilde{x}\|_2$, and hence the convex inequality constraint (3.19) defines the halfspace opposite to the unit ball with respect to this hyperplane. The dotted region in figure 3.9 shows the corresponding feasible set of (3.19) in $\mathbf{R}^2$ with $\tilde{x} = (3, 1)$. It is clear in the figure that the feasible set defined by this convex restriction is a subset of the original feasible set in (3.17).

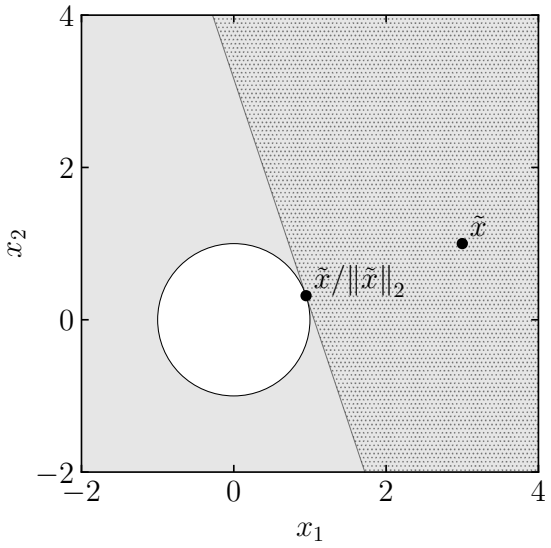

Figure 3.9 *Convex restrictions in CCP.* The shaded region shows the feasible set of the nonconvex constraint $\|x\|_2 \geq 1$ in $\mathbf{R}^2$. In each CCP iteration, this constraint is replaced by a convex restriction $(1/\|\tilde{x}\|_2)\tilde{x}^T x \geq 1$ at the current point $\tilde{x}$. The dotted region shows an example of the corresponding restricted feasible set with $\tilde{x} = (3, 1)$.

Algorithm

The full algorithm of CCP is summarized as follows.

Algorithm 3.2 CONVEX-CONCAVE PROCEDURE.

given a starting point $x^{(0)}$ feasible to (3.12).

$k := 0$.

repeat

1. *Convexify.* Form (3.16) with $\tilde{x} = x^{(k)}$ for $i = 0, \ldots, m$.

$$\hat{g}_{i,x^{(k)}}(x) := g_i(x^{(k)}) + \nabla g_i(x^{(k)})^T(x - x^{(k)}), \quad i = 0, \ldots, m.$$

2. *Solve the convex approximation (3.15).*

$$x^{(k+1)} := \operatorname*{argmin}_{x \in \mathbf{R}^n} \left\{ f_0(x) - \hat{g}_{0,x^{(k)}}(x) \;\middle|\; f_i(x) - \hat{g}_{i,x^{(k)}}(x) \leq 0,\ i = 1, \ldots, m \right\}.$$

3. $k := k + 1$.

until stopping criteria is satisfied.

One reasonable choice of stopping criteria is to terminate the CCP iterations when the change of objective values between two consecutive iterations is below

some small threshold $\epsilon > 0$, *i.e.*, quit when

$$|(f_0(x^{(k+1)}) - g_0(x^{(k+1)})) - (f_0(x^{(k)}) - g_0(x^{(k)}))| < \epsilon.$$

Some other options including stopping when the change of variable values is below some threshold, *i.e.*, quit when

$$\|x^{(k+1)} - x^{(k)}\|_2 < \epsilon,$$

or simply limiting the maximum number of CCP iterations. Also note that, again, as a heuristic method, algorithm 3.2 is sensitive to initialization.

It can be shown (under some technical conditions) that the sequence of points $\{x^{(k)}\}_{k=0}^{\infty}$ obtained from algorithm 3.2 is always feasible to the original problem (3.12), and that the sequence of objective values $\{f_0(x^{(k)}) - g_0(x^{(k)})\}_{k=0}^{\infty}$ is monotonically nonincreasing (see exercise 3.2). Furthermore, if the sequence $\{x^{(k)}\}_{k=0}^{\infty}$ converges to some point $x^\star$, then $x^\star$ is a stationary point (which, note that, need not be a local minimum) of the objective function of (3.12) over the corresponding feasible set.

3.2.4 Numerical examples

In this section we illustrate the basic idea and properties of CCP with some numerical examples.

Minimizing a difference-of-convex function

We first consider an unconstrained minimization problem of the difference-of-convex function $h\colon \mathbf{R} \to \mathbf{R}$ given by

$$\text{minimize} \quad h(x) = f(x) - g(x) = x^4 + (1/5)x - x^2, \tag{3.20}$$

where the convex components are given by $f(x) = x^4 + (1/5)x$ and $g(x) = x^2$. In each CCP iteration, we minimize over its first-order majorizer $\hat{h}_{\tilde{x}}\colon \mathbf{R} \to \mathbf{R}$ at the current point $\tilde{x}$ obtained from linearizing $g(x) = x^2$ according to (3.16), *i.e.*,

$$\text{minimize} \quad \hat{h}_{\tilde{x}}(x) = f(x) - \hat{g}_{\tilde{x}}(x) = x^4 + (1/5)x - 2\tilde{x}x + \tilde{x}^2.$$

Figures 3.10 and 3.11 show two example runs of CCP (the initial 4 iterations) on the problem (3.20), starting from different initial points, *i.e.*, $x^{(0)} = -0.2$ and $x^{(0)} = 0.2$, respectively. It is clear from the figures that the first-order majorizers $\hat{h}_{\tilde{x}}$ (shown dashed) at different CCP iterations are always a convex global upper bound of the original objective function h (shown solid), and are tight at the current point (shown circle). In both cases, the sequence of optimal points (shown in stars) from minimizing these majorizers at each CCP iteration leads to a sequence of objective values that is monotonically nonincreasing, but from different initial points, the CCP iterations converge to different stationary points of the original objective h. In particular, when starting from $x^{(0)} = -0.2$, the final convergent point is the global minimum of h, while starting from $x^{(0)} = 0.2$ leads to a locally optimal

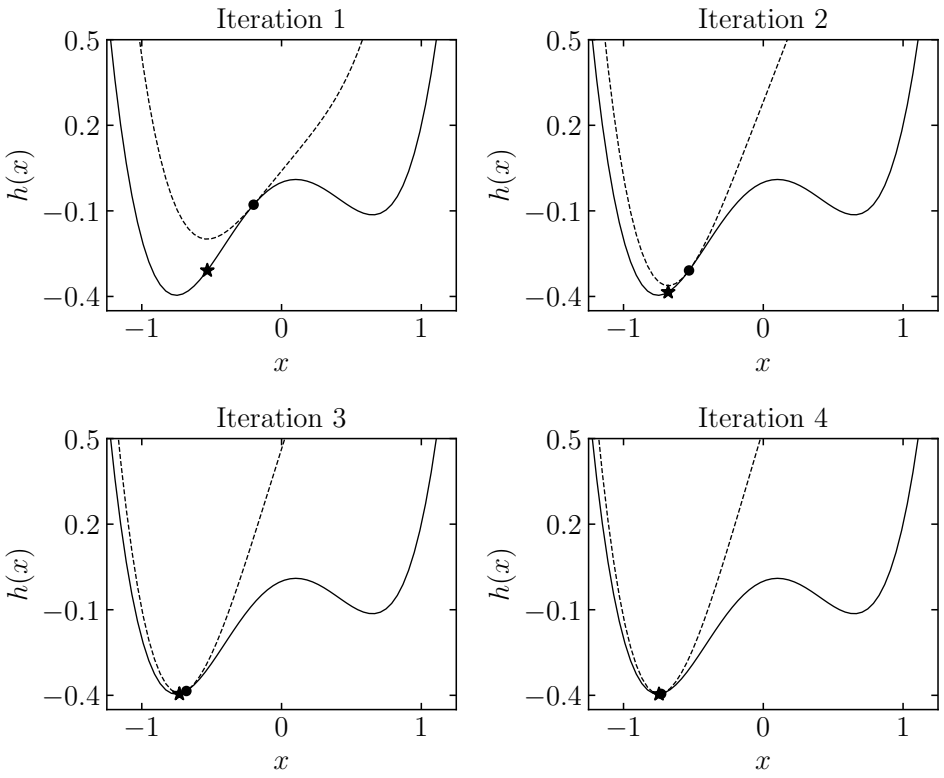

Figure 3.10 First 4 CCP iterations for the problem (3.20) starting from $x^{(0)} = -0.2$. The original objective function h and its first-order majorizers at each CCP iteration are shown solid and dashed, respectively. The current point at each CCP iteration is shown circle, and the optimal point of each majorizer (evaluated at the original objective h) is shown star. The final convergent point is the global minimum of h.

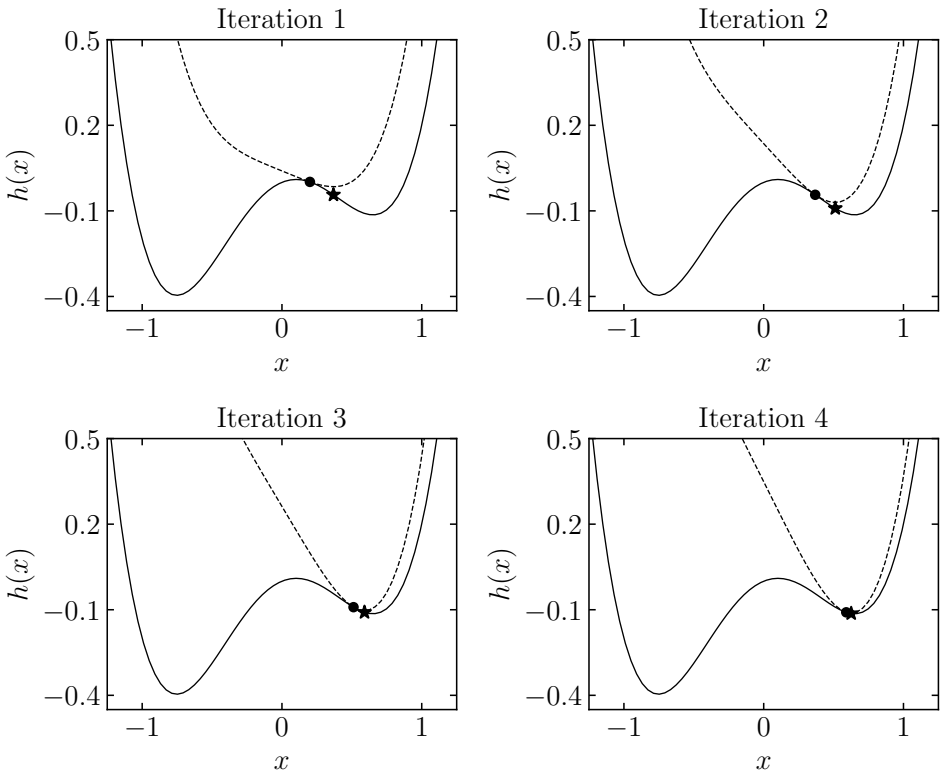

Figure 3.11 First 4 CCP iterations for the problem (3.20) starting from $x^{(0)} = 0.2$. The original objective function h and its first-order majorizers at each CCP iteration are shown solid and dashed, respectively. The current point at each CCP iteration is shown circle, and the optimal point of each majorizer (evaluated at the original objective h) is shown star. The final convergent point is locally optimal but not globally optimal.

but not globally optimal convergent point. This observation shows that the final convergent point of CCP is indeed dependent on the initialization.

In this specific example of the difference-of-convex problem (3.20), the CCP iterations do converge to some local minimum as shown in figures 3.10 and 3.11, but this is not always the case in general. For example, consider minimizing the difference-of-convex function $h(x) = x^4 - x^2$ on $\mathbf{R}$. If the initial point is chosen to be $x^{(0)} = 0$, then the CCP iterations will converge immediately after one step to the stationary point $x^\star = 0$, which is actually a local maximum of h.

Closest point outside unit ball

As another example, consider the problem of finding the closest point to a given point $c \in \mathbf{R}^n$ within the unit Euclidean ball centered at the origin, from the outside of the unit ball, *i.e.*,

$$\begin{array}{ll} \text{minimize} & \|x - c\|_2 \\ \text{subject to} & \|x\|_2 \geq 1, \end{array} \tag{3.21}$$

where $x \in \mathbf{R}^n$ is the optimization variable. Although the objective of (3.21) is convex, the constraint $\|x\|_2 \geq 1$ is not, but nevertheless can be expressed in the difference-of-convex form as $0 - (\|x\|_2 - 1) \leq 0$. Thus, in each CCP iteration, we solve the following convex approximation of (3.21):

$$\begin{array}{ll} \text{minimize} & \|x - c\|_2 \\ \text{subject to} & (1/\|\tilde{x}\|_2)\tilde{x}^T x \geq 1, \end{array}$$

where $\tilde{x}$ is the current point (see example 3.8).

Figure 3.12 shows an example run of CCP (the initial 4 iterations) on the problem (3.21) in $\mathbf{R}^2$ with $c = (-0.4, 0.6)$ (shown cross), starting from the initial point $x^{(0)} = (1.9, 1.5)$. Geometrically, the CCP iterations iteratively generate a sequence of convex restrictions of the original feasible set (which is a halfspace defined by a tangent hyperplane of the unit ball, shown as the dotted area), and find a point within this convex restriction that is closest to the target point c. It is observed from the figure that after 4 CCP iterations, the current point $x^{(4)}$ is already very close to the optimal point of the original problem (3.21).

Figure 3.13 shows the objective values of (3.21) at each CCP iteration with different initial points. The curve corresponds to the initial point in figure 3.12 is shown thicker. Although different initial points lead to different convergence paths, in all cases the CCP iterations successfully find the global optimum of the problem (3.21).

3.3 General nonlinear optimization

In this section we consider the most general form of nonlinear programming problems, and introduce the generic *sequential convex approximation* (SCA) heuristic for approximately solving them. The ACS and CCP heuristic introduced previously can be viewed as special cases of SCA applied to biconvex optimization problems and

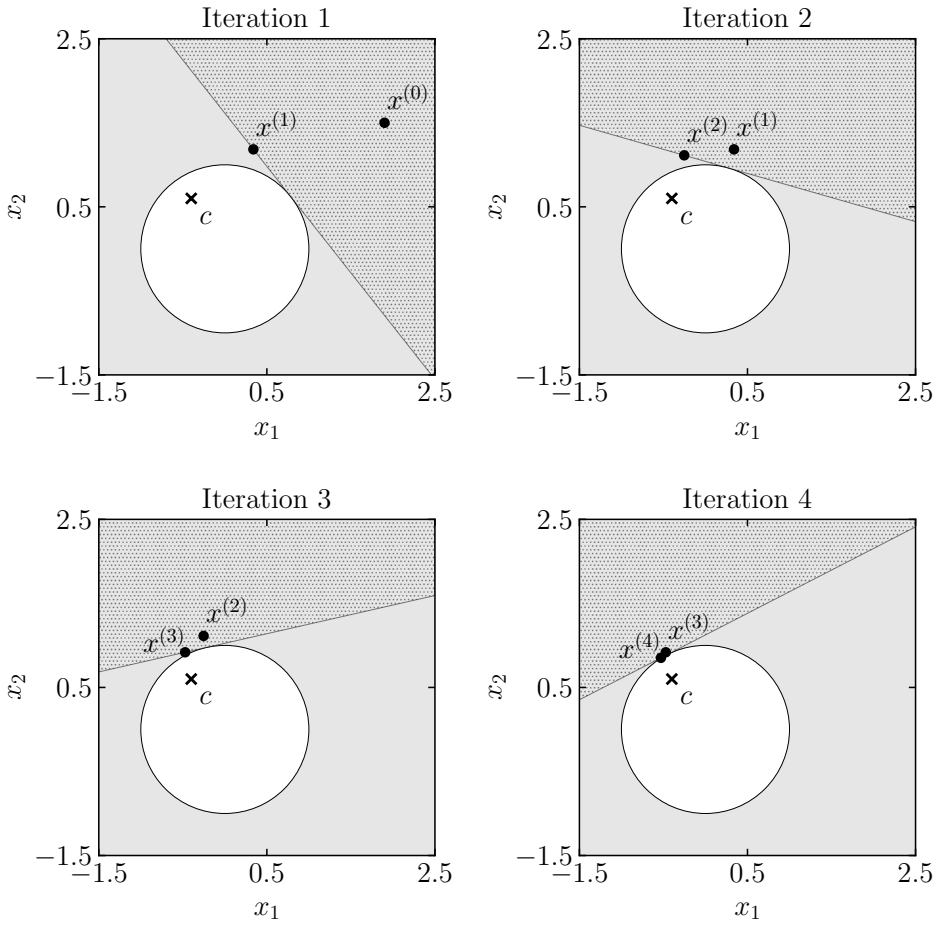

Figure 3.12 First 4 CCP iterations for the problem (3.21) in $\mathbf{R}^2$ with $c = (-0.4, 0.6)$ (shown cross), starting from the initial point $x^{(0)} = (1.9, 1.5)$. The feasible set of (3.21) is the region outside the unit ball (shown shaded). The dotted region corresponds to the convex restriction of the feasible set for each CCP iteration.

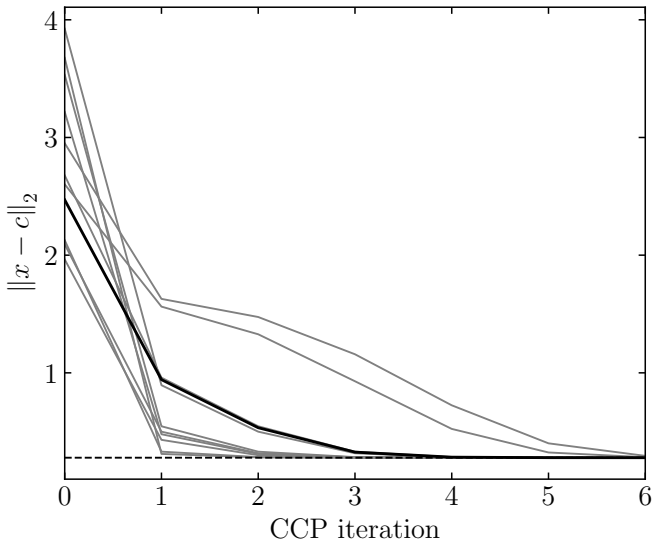

Figure 3.13 Objective values of the problem (3.21) at each CCP iteration with different initial points. The curve corresponding to the initial point in figure 3.12 is shown thicker. The dashed line shows the (globally) optimal objective value of (3.21).

difference-of-convex programming problems, respectively. The SCA method is applicable to a much broader class of problems, but it is in general computationally more expensive and involves more technical issues which might vary wildly across different applications, so implementing a SCA algorithm in practice are more or less a combination of both art and technology. Here we only focus on the basic ideas of SCA, and leave (some generally appeared) associated technical issues to §C.3 in the appendix.

3.3.1 Sequential convex approximation

Suppose we are given the following optimization problem:

$$\begin{array}{lll} \text{minimize} & f_0(x) & \\ \text{subject to} & f_i(x) \leq 0, & i = 1, \ldots, m \\ & h_i(x) = 0, & i = 1, \ldots, p, \end{array} \tag{3.22}$$

where $x \in \mathbf{R}^n$ is the optimization variable, and the functions $f_i \colon \mathbf{R}^n \to \mathbf{R}$ for $i = 0, \ldots, m$ are possibly nonconvex and $h_i \colon \mathbf{R}^n \to \mathbf{R}$ for $i = 1, \ldots, p$ are possibly nonaffine.

The basic idea of SCA is to iteratively form a convex approximation of the functions f_i, $i = 0, \ldots, m$, and an affine approximation of the equality constraint functions h_i, $i = 1, \ldots, p$, over some *trust region* $\mathcal{T} \subseteq \mathbf{R}^n$ around the current point, and then solve the resulting convex optimization problem to update the variable

values. (Therefore, SCA is sometimes referred to as the *trust region methods*.) Specifically, let $\tilde{x} \in \mathbf{R}^n$ be the current value of the problem variable, then in the next SCA iteration, we solve the following approximation of the original problem:

$$\begin{array}{ll} \text{minimize} & \hat{f}_{0,\tilde{x}}(x) \\ \text{subject to} & \hat{f}_{i,\tilde{x}}(x) \leq 0, \quad i = 1, \ldots, m \\ & \hat{h}_{i,\tilde{x}}(x) = 0, \quad i = 1, \ldots, p \\ & x \in \mathcal{T}_{\tilde{x}}, \end{array} \tag{3.23}$$

where $\hat{f}_{i,\tilde{x}} \colon \mathbf{R}^n \to \mathbf{R}$ for $i = 0, \ldots, m$ are the convex approximations of f_i and $\hat{h}_{i,\tilde{x}} \colon \mathbf{R}^n \to \mathbf{R}$ for $i = 1, \ldots, p$ are the affine approximations of h_i, over the current trust region $\mathcal{T}_{\tilde{x}}$. The additional trust region constraint in (3.23) can be interpreted as a restriction on the next point enforcing it not to be too far away from the current point $\tilde{x}$, so that the convex and affine approximations $\hat{f}_{i,\tilde{x}}$ and $\hat{h}_{i,\tilde{x}}$ are more likely to be close to the original functions f_i and h_i, respectively. Typically, the trust region $\mathcal{T}_{\tilde{x}}$ is chosen as a convex set containing the current point $\tilde{x}$, so that the problem (3.23) is a convex optimization problem and hence can be efficiently solved.

In general, the trust region $\mathcal{T}_{\tilde{x}}$ is defined as either a Euclidean ball with center $\tilde{x}$ and radius $\rho > 0$, *i.e.*,

$$\mathcal{T}_{\tilde{x}} = \{x \in \mathbf{R}^n \mid \|x - \tilde{x}\|_2 \leq \rho\},$$

or a box centered at $\tilde{x}$, *i.e.*,

$$\mathcal{T}_{\tilde{x}} = \{x \in \mathbf{R}^n \mid |x_i - \tilde{x}_i| \leq \rho_i,\ i = 1, \ldots, n\},$$

where $\rho \in \mathbf{R}^n_{++}$ is a vector of positive scalars that defines the size of the trust region for each component of x. In the second case, if x_i appears only in the convex objective, convex inequality constraints, and affine equality constraints, then we can set $\rho = \infty$ for that component, so that no trust region is imposed on it.

3.3.2 Convexification methods

We now present some widely used methods for forming convex and affine approximations of some general nonconvex functions f_i and h_i in (3.22).

Taylor expansions

Probably the first idea appeared in mind when talking about forming convex approximations of some functions is to use their Taylor expansions around the current point.

In particular, suppose we are given a twice differentiable function $f \colon \mathbf{R}^n \to \mathbf{R}$, then an affine approximation of f at some point $\tilde{x} \in \mathbf{R}^n$ is given by the first-order Taylor expansion:

$$\hat{f}_{\tilde{x}}(x) = f(\tilde{x}) + \nabla f(\tilde{x})^T (x - \tilde{x}). \tag{3.24}$$

An example of this idea in $\mathbf{R}$ is illustrated in figure 3.14.

A convex approximation of f at $\tilde{x}$ can be obtained from the second-order Taylor expansion by taking only the positive semidefinite part of its Hessian $\nabla^2 f(\tilde{x})$.

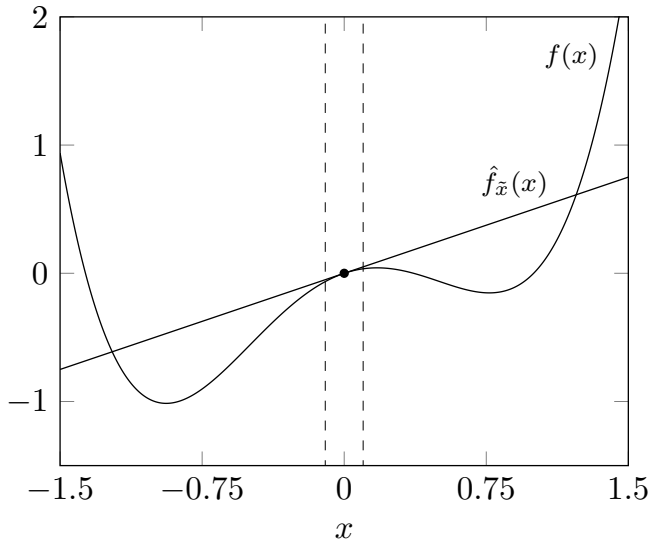

Figure 3.14 An affine approximation in the form (3.24) of the nonconvex function $f(x) = x^4 + (1/2)x - (3/2)x^2$ at the point $\tilde{x} = 0$ (shown circle). The boundary of a trust region around $\tilde{x}$ is shown dashed.

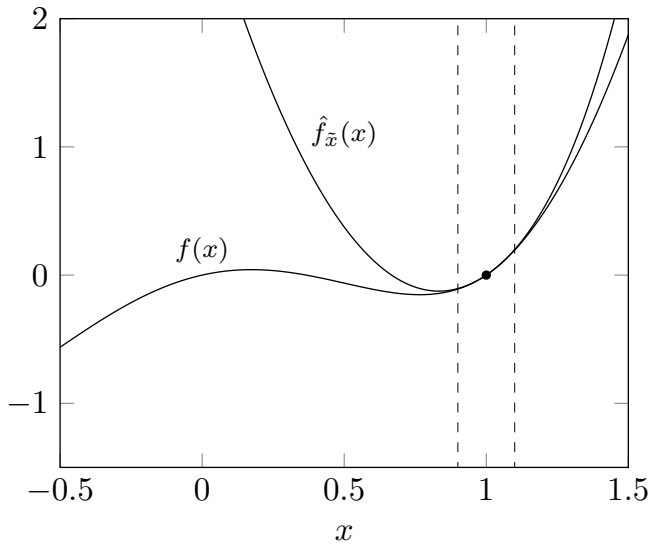

Figure 3.15 A convex approximation in the form (3.25) of the nonconvex function $f(x) = x^4 + (1/2)x - (3/2)x^2$ at the point $\tilde{x} = 1$ (shown circle). The approximation $\hat{f}_{\tilde{x}}$ is a convex quadratic function, and is very close to f within a small trust region around $\tilde{x}$ (whose boundary is shown dashed).

Specifically, let $\nabla^2 f(\tilde{x}) = Q\Lambda Q^T$ be the eigenvalue decomposition of $\nabla^2 f(\tilde{x})$, where Q is orthogonal and $\Lambda = \mathbf{diag}(\lambda_1, \ldots, \lambda_n)$ is a diagonal matrix whose diagonal elements are the eigenvalues of $\nabla^2 f(\tilde{x})$. The matrix $H_+ = Q\Lambda_+ Q^T$ with

$$\Lambda_+ = \mathbf{diag}(\max\{\lambda_1, 0\}, \ldots, \max\{\lambda_n, 0\})$$

is therefore the positive semidefinite part of $\nabla^2 f(\tilde{x})$. The (second-order) convex approximation of f at $\tilde{x}$ is then given by

$$\hat{f}_{\tilde{x}}(x) = f(\tilde{x}) + \nabla f(\tilde{x})^T (x - \tilde{x}) + (1/2)(x - \tilde{x})^T H_+ (x - \tilde{x}). \tag{3.25}$$

Notice that the convex approximation $\hat{f}_{\tilde{x}}$ of the function f obtained from the second-order Taylor expansion (3.25) are typically convex quadratic functions, so SCA with such convexification methods are also called *sequential quadratic programming*. Figure 3.15 shows an example of the approximation (3.25) in $\mathbf{R}$.

Convex and affine approximations established from Taylor expansions are sometimes called *local models*, since if the trust region is restricted to be small around the current point $\tilde{x}$, then the difference between the original function f and its approximation $\hat{f}_{\tilde{x}}$ within this region could be very small. On the other hand, if the trust region is large, then the approximation $\hat{f}_{\tilde{x}}$ might deviate significantly from the original function f within this region. These properties are also shown in the figures 3.14 and 3.15. From these ideas it is hence clear that the size of the trust region $\mathcal{T}_{\tilde{x}}$ in (3.23) should not be too large when local convexification methods are used, since otherwise such local approximations might be very inaccurate.

Particle methods

A modern approach of forming convex and affine approximations of general nonconvex functions is to use *particle methods*, which require essentially no assumptions on the original functions except that they can be evaluated at given points.

Suppose we would like to obtain some convex or affine approximation to the function $f\colon \mathbf{R}^n \to \mathbf{R}$ around the point $\tilde{x}$, the basic idea of particle methods is to sample a set of points $z_1, \ldots, z_K \in \mathcal{T}_{\tilde{x}}$ within the trust region and evaluate f at these points to obtain $y_i = f(z_i)$ for $i = 1, \ldots, K$, so that we have a set of *particles* (z_i, y_i), $i = 1, \ldots, K$. These particles may be sampled randomly within the trust region, or generated from some deterministic scheme, such as using a grid, or from taking the extreme points of the trust region. Then convex and affine approximations of f over the trust region $\mathcal{T}_{\tilde{x}}$ can be formed by fitting a convex or an affine function to these particles, using convex optimization. (The ideas of fitting and approximation will be presented more formally in §4.1.)

The advantages of particle methods include that they allow for general nondifferentiable functions, or functions for which evaluating derivatives is very challenging. More importantly, particle methods give *regional models* that are accurate over the whole trust region $\mathcal{T}_{\tilde{x}}$ (assuming enough coverage of $\mathcal{T}_{\tilde{x}}$ in the evaluated points $z_1, \ldots, z_K$), rather than only locally around the current point $\tilde{x}$. In other words, accuracy of the convex approximation obtained from particle methods is not sensitive to the size of the trust region $\mathcal{T}_{\tilde{x}}$, so larger trust regions can be used in SCA with particle methods. On the other hand, particle methods typically have

extraordinary sampling and function evaluation requirements, since in general the number of particles required to cover the trust region $\mathcal{T}_{\tilde{x}}$ grows exponentially with the dimension of the variables n.

Some examples illustrate these ideas.

Example 3.9 *Fitting affine and convex quadratic approximations.* Suppose we would like to form an affine approximation of the function $f\colon \mathbf{R}^n \to \mathbf{R}$ around the point $\tilde{x} \in \mathbf{R}^n$ over the trust region $\mathcal{T}_{\tilde{x}}$, and let $(z_i, y_i) = (z_i, f(z_i))$ for $i = 1, \ldots, K$ be the set of particles sampled within $\mathcal{T}_{\tilde{x}}$. An affine approximation of f at $\tilde{x}$ in the form

$$\hat{f}_{\tilde{x}}(x) = a^T(x - \tilde{x}) + b$$

can be obtained by solving the following least squares problem:

$$\text{minimize} \quad \sum_{i=1}^{K} \left(a^T(z_i - \tilde{x}) + b - y_i\right)^2$$

with variables $a \in \mathbf{R}^n$ and $b \in \mathbf{R}$.

Figure 3.16 shows an example in $\mathbf{R}$. The function

$$f = x^4 + (1/2)x - (3/2)x^2 \tag{3.26}$$

is sampled to generate 10 particles (shown cross) within the trust region $\mathcal{T}_{\tilde{x}} = \{x \mid |x| \leq 0.5\}$ around the point $\tilde{x} = 0$. The fitted affine approximation $\hat{f}_{\tilde{x}}$ of the function f according to these particles is shown thicker. The dashdotted line plots the first-order Taylor expansion of f at $\tilde{x} = 0$ for comparison. It is clear from the figure that although the local model from Taylor expansion is very accurate around the current point $\tilde{x}$, the regional model from particle fitting on average aligns better with the original function f over the whole trust region.

Similar idea can be used to fit convex quadratic approximations of f. Consider a convex quadratic function of the form

$$\hat{f}_{\tilde{x}}(x) = (1/2)(x - \tilde{x})^T P(x - \tilde{x}) + q^T(x - \tilde{x}) + r, \tag{3.27}$$

where $P \in \mathbf{S}^n_+$, $q \in \mathbf{R}^n$, and $r \in \mathbf{R}$ are the parameters to be determined. A convex approximation of f in the form (3.27) at $\tilde{x}$ can then be obtained by solving

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{K} \left((1/2)(z_i - \tilde{x})^T P(z_i - \tilde{x}) + q^T(z_i - \tilde{x}) + r - y_i\right)^2 \\ \text{subject to} & P \succeq 0 \end{array}$$

with variables $P \in \mathbf{S}^n$, $q \in \mathbf{R}^n$, and $r \in \mathbf{R}$, which is a constrained least squares problem (and hence convex).

Figure 3.17 shows an example in $\mathbf{R}$. We consider again the function f given by (3.26) and sample 20 particles (shown cross) within the trust region $\mathcal{T}_{\tilde{x}} = \{x \mid |x - 1| \leq 0.5\}$ around the point $\tilde{x} = 1$. It is observed in the figure that the fitted convex quadratic approximation (3.27) on these particles in general provides a good regional model of the function f over the whole trust region, while the local model from Taylor expansion (shown dashdotted) deviates significantly from f when getting close to the boundary of $\mathcal{T}_{\tilde{x}}$.

Example 3.10 *Fitting convex envelopes.* Suppose we are given a function $f\colon \mathbf{R}^n \to \mathbf{R}$ and some sampled particles (z_i, y_i), $i = 1, \ldots, K$, within some trust region $\mathcal{T}$, where

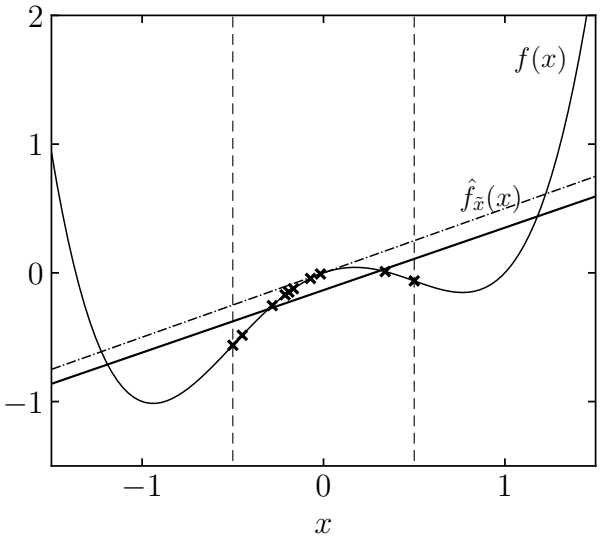

Figure 3.16 An affine approximation of the function (3.26) at the point $\tilde{x} = 0$ obtained from fitting to 10 particles (shown cross) in $\mathcal{T}_{\tilde{x}} = \{x \mid |x| \leq 0.5\}$. The approximation $\hat{f}_{\tilde{x}}$ from particle fitting is shown thicker. The first-order Taylor expansion of f at $\tilde{x} = 0$ is shown dashdotted for reference.

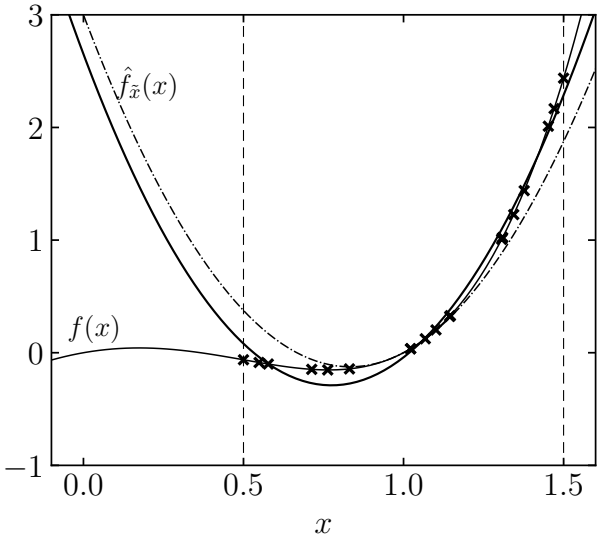

Figure 3.17 A convex quadratic approximation of the function (3.26) at $\tilde{x} = 1$ from fitting to 20 particles (shown cross) in $\mathcal{T}_{\tilde{x}} = \{x \mid |x - 1| \leq 0.5\}$. The approximation $\hat{f}_{\tilde{x}}$ from particle fitting is shown thicker. The second-order Taylor expansion of f at $\tilde{x} = 1$ is shown dashdotted.

$y_i = f(z_i)$ for all $i = 1, \ldots, K$. To form a convex approximation of f based on these particles, we may consider fitting a convex lower bound that is as close to the particles as possible, which is in effect approximately taking the *convex envelope* (see page 38) of the function f over $\mathcal{T}$. There are many ways to do this; here we present one simple approach.

Consider a piecewise affine function $h\colon \mathbf{R}^n \to \mathbf{R}$ of the form

$$h(x) = \max\{h_i + g_i^T(x - z_i) \mid i = 1, \ldots, K\}, \tag{3.28}$$

where $g_i \in \mathbf{R}^n$ and $h_i \in \mathbf{R}$ for $i = 1, \ldots, K$ are the parameters to be determined. The function h is clearly convex as it is the pointwise maximum of a set of affine functions. To make h a lower bound of f over the particles (z_i, y_i), $i = 1, \ldots, K$, it has to satisfy

$$y_i \geq h(z_i) = \max\{h_j + g_j^T(z_i - z_j) \mid j = 1, \ldots, K\}$$

for all $i = 1, \ldots, K$, which is equivalent to

$$y_i \geq h_j + g_j^T(z_i - z_j)$$

for all $i, j = 1, \ldots, K$. Noticing that when evaluating the ith piece of the function h at $x = z_i$, we have

$$h_i + g_i^T(z_i - z_i) = h_i,$$

so the value of h at the point z_i is at least h_i, *i.e.*,

$$h(z_i) = \max\{h_j + g_j^T(z_i - z_j) \mid j = 1, \ldots, K\} \geq h_i.$$

Therefore, to make h as close to the particles as possible from below, in the sense of maximizing the sum of the values of h at all particles, *i.e.*, maximizing $\sum_{i=1}^K h(z_i)$, we can equivalently maximize the sum of h_i for $i = 1, \ldots, K$. Putting these together, to fit a convex approximation of f in the form (3.28) we can solve the following optimization problem:

$$\begin{array}{ll} \text{maximize} & \sum_{i=1}^K h_i \\ \text{subject to} & y_i \geq h_j + g_j^T(z_i - z_j), \quad i,\ j = 1, \ldots, K \end{array} \tag{3.29}$$

with variables $g_i \in \mathbf{R}^n$ and $h_i \in \mathbf{R}$ for $i = 1, \ldots, K$, which is simply a linear program.

Figure 3.18 shows an example to illustrate this approach's performance, where the target function $f\colon \mathbf{R} \to \mathbf{R}$ is given by (3.26). Each subplot shows the convex envelope approximation of f obtained from solving (3.29) on different numbers of particles (shown cross) sampled with equal spacing from the trust region $\mathcal{T} = \{x \mid |x| \leq 1.5\}$. It is clear from the figure that the fitted piecewise affine functions in the form (3.28) are always a convex lower bound of the particles, and as the number of particles increases, the fitted convex envelope approximation gets closer to the actual convex envelope of f over the trust region $\mathcal{T}$.

3.3.3 Numerical example

As a basic numerical example of SCA applications, we consider the following non-convex quadratic program:

$$\begin{array}{ll} \text{minimize} & f(x) = (1/2)x^T P x + q^T x \\ \text{subject to} & \|x\|_\infty \leq 1 \end{array} \tag{3.30}$$

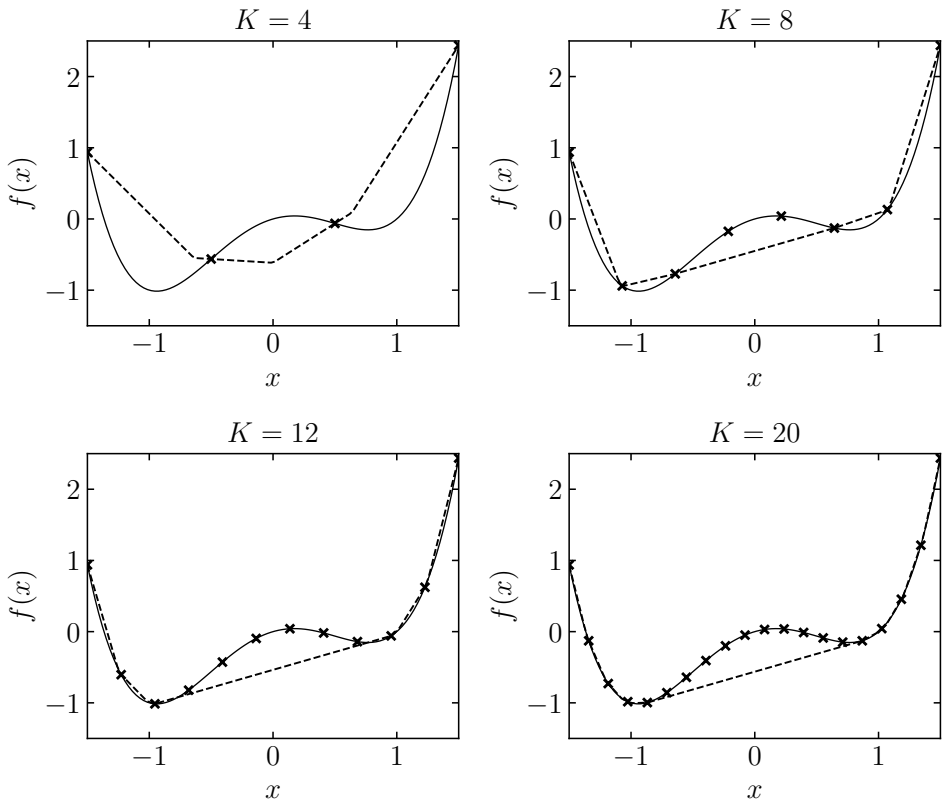

Figure 3.18 Convex envelope approximations of the function (3.26) over the trust region $\mathcal{T} = \{x \mid |x| \leq 1.5\}$ obtained from solving (3.29) on different numbers of particles (shown cross). The target function f is shown solid and the convex approximation is shown dashed.

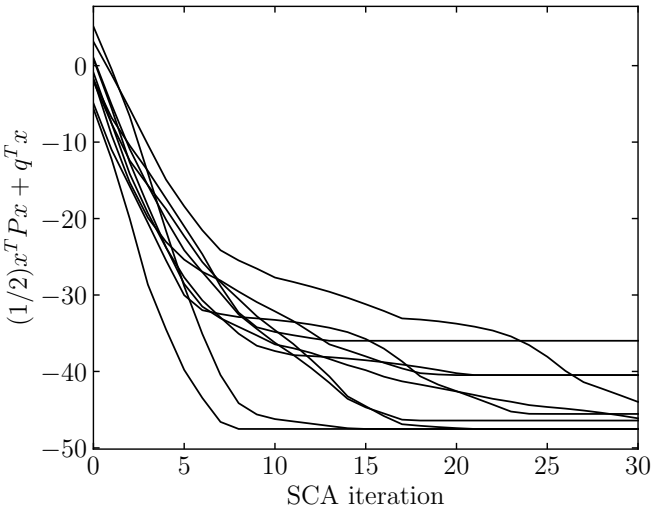

Figure 3.19 Convergence behavior of SCA applied to the nonconvex quadratic program (3.30) from 10 different initial points.

with variable $x \in \mathbf{R}^n$, where the matrix $P \in \mathbf{S}^n$ is symmetric but possibly indefinite. In this case, we may write the second-order Taylor expansion (3.25) of the objective function f at the current point $\tilde{x}$ as

$$\hat{f}_{\tilde{x}}(x) = f(\tilde{x}) + (P\tilde{x} + q)^T(x - \tilde{x}) + (1/2)(x - \tilde{x})^T P_+(x - \tilde{x}),$$

where P_+ is the projection of P onto the positive semidefinite cone $\mathbf{S}^n_+$. Therefore, in each SCA iteration, we solve the following convex quadratic approximation of the original problem (3.30):

$$\begin{array}{ll} \text{minimize} & f(\tilde{x}) + (P\tilde{x} + q)^T(x - \tilde{x}) + (1/2)(x - \tilde{x})^T P_+(x - \tilde{x}) \\ \text{subject to} & \|x\|_\infty \leq 1, \quad x \in \mathcal{T}_{\tilde{x}}, \end{array}$$

where $x \in \mathbf{R}^n$ is the variable and $\tilde{x}$ is the current point. The trust region $\mathcal{T}_{\tilde{x}}$ is chosen as the box

$$\mathcal{T}_{\tilde{x}} = \{x \in \mathbf{R}^n \mid \|x - \tilde{x}\|_\infty \leq \rho\}$$

with some $\rho > 0$.

We try to solve an instance of the problem (3.30) with $n = 20$ via SCA, where the matrix P and the vector q generated randomly. The SCA procedure is initialized from 10 different random points with $\|x\|_\infty \leq 1$, and the trust region size is set to $\rho = 0.2$ for all iterations. Figure 3.19 shows the objective values at each SCA iteration for all 10 initial points.

There are several observations from the results. Firstly, the SCA procedure from most of the 10 initial points converges within 30 iterations. On the other hand, the final point returned by SCA is substantially influenced by the initial points, and some initialization can lead to very poor final objective values. Most importantly,

although some initial points lead to quite small final objective values, it is never clear whether these points have solved the original problem (3.30) or not, and we also have no information about how far they are from the global optimum. We should also note that in this specific example, the objective values at all SCA iterates are nonincreasing, which is not necessarily true in general SCA applications, especially when the original problem involves many nonconvex constraints.

Bibliographical notes

The first notice of biconvexity structure in the context of mathematical programming can be traced back to Falk *et al.* [FS69] in the 1960s. Most results on the analysis of biconvex sets can be found in the papers of Aumann and Hart [AH86] and Goh *et al.* [GTS+94]. Properties of biconvex functions that are most relevant to optimization problems can be found in Goh *et al.* [GTS+94] and Gorski *et al.* [GPK07].

Biconvex problems appears in various application domains; see [GPK07] and [ZB25] for a survey and references. Toker and Özbay [TÖ95] showed that solving bilinear matrix inequalities is NP-hard. Some useful properties of partial optimality conditions (*i.e.*, conditions for stationary points) of biconvex problems can be found in [WH76, GPK07]. The necessary conditions for a partially optimal point of a biconvex problem being a local optimum are discussed in [GPK07], but in general, no stronger results can be obtained.

ACS methods are a special case of *block relaxation methods* (which is also known as *block coordinate descent* methods); see [War63, Pow73, De 94]. A survey on ACS methods for biconvex problems can be found in [WH76]. Gorski *et al.* [GPK07] showed that under weak assumptions all solution points generated by ACS form a compact connected set and that each of these points is a stationary point of the objective function. However, there is currently no better convergence results regarding local or global optimality properties.

The idea of ACS can be readily extended to *multiconvex optimization problems*, which involve functions that are convex in each of more than two blocks of variables when the other blocks are fixed; see [SDU+17].

The broad class of difference-of-convex functions includes all twice continuously differentiable functions [Har59], so the difference-of-convex programming problems (3.12) is very general. Many problems that are widely believed to be very hard, *e.g.*, the traveling salesman problem, can be represented in the form of the boolean least squares problem (3.13) [Kar72].

Difference-of-convex problems have been studied for several decades; some early works include [Tuy86, TH88, HPTD91]. Early researches on this topic mainly focus on solving difference-of-convex problems globally, where most of the algorithms rely on *branch-and-bound* or *cutting-plane* methods as in [MF97]. Good overviews for solving difference-of-convex problems globally can be found in [HT99, HPT00].

The CCP heuristic for attempting to solve difference-of-convex problems was first proposed geometrically by Yuille and Rangarajan [YR03], but without inequality constraints. The ideas of considering CCP as a sequence of convex approximations of the original problem and adding inequality constraints were later discussed by Smola *et al.* [SVH05]. A more recent overview of CCP and its applications is presented by Lipp and Boyd [LB16]. Sriperumbudur and Lanckriet [SL09] discussed the convergence properties of CCP.

Throughout the discussions on difference-of-convex problems and the CCP heuristic in this chapter, we have assumed that the involved functions are differentiable. In fact, CCP can also be applied to nondifferentiable problems, by replacing the gradients in (3.16) with *subgradients*; see, *e.g.*, [LB16] and [SDGB16] for more details. Extensions of CCP to handle generalized inequality constraints, as well as its combination with cutting-plane methods for large-scale difference-of-convex problems which involve a large number of variables and constraints, are presented in [LB16].

ACS and CCP are sometimes considered as special cases of the *majorization-minimization* (MM) algorithm [LHY00], in which a minimization problem is approximated by an easier

to solve upper bound created around the current point (*i.e.*, majorization) and then minimized. The most famous example of MM is the *expectation-maximization* (EM) algorithm, which was introduced in [DLR77]. Although MM can be first traced back to the 1970s in [De 05], the term majorization-minimization was not coined until several decades later in [LHY00]. Many MM extensions have been developed over the years and more discussions on these algorithms can be found in [MK07, Lan13, LR19].

Sequential convex approximation methods are most widely known under the name *sequential quadratic programming*, while it is, strictly speaking, a subset of SCA methods that use convex quadratic approximations via Taylor expansion. Sequential quadratic programming was first proposed by Wilson [Wil63] in the 1960s for solving nonlinear programming problems, and its convergence properties were later shown by Robinson [Rob72]. There are numerous references on SCA methods and related subtopics, and most of which are classical optimization textbooks, *e.g.*, Boggs and Tolle [BT95], Conn *et al.* [CGT00], Nocedal and Wright [NW06], Gill and Wong [GW12], and Bertsekas [Ber16].

A useful extension of SCA methods was originally introduced by Griewank [Gri81] in an unpublished technical note, which was later rediscovered independently by Nesterov and Polyak [NP06], and Weiser *et al.* [WDE07]. They consider applying SCA to some unconstrained nonlinear program, where in each iteration, a *cubic* regularization in the form $(\rho/3)\|x - \tilde{x}\|_2^3$ on the variable x is added as an additional regularization term to the quadratically approximated objective. This regularized SCA method was shown to have some favorable convergence properties, and the ideas were later unified and extended into a coherent and numerically efficient algorithmic framework by Cartis *et al.* [CGT11a, CGT11b].

Another convexification method that can be applied in SCA involves *convex composite* functions; see [FW80, Fle82, Bur85, Fle87, BF95, DL18] for more details. For general convex function fitting and interpolation via particle methods, see [BV04, §6.5].

Exercises

3.1 *Difference-of-convex decomposition.* Suppose $f : \mathbf{R}^n \to \mathbf{R}$ is twice differentiable, and there exists some constant $M \geq 0$ such that

$$\nabla^2 f(x) \succeq -MI$$

for all $x \in \mathbf{R}^n$. Show that f can be expressed as the difference of two convex functions.

3.2 *Convergence of convex-concave procedure.* Show that the sequence of points $\{x^{(k)}\}_{k=0}^{\infty}$ obtained from the CCP iterations (algorithm 3.2) is always feasible to the original difference-of-convex problem (3.12), and that the corresponding sequence of objective values, given by $\{f_0(x^{(k)}) - g_0(x^{(k)})\}_{k=0}^{\infty}$, is monotonically nonincreasing.

Part II

Disciplined modules

Chapter 4

Objectives

4.1 Approximation

4.1.1 Residuals

Consider a set of m (possibly nonlinear) equations in n variables $x \in \mathbf{R}^n$, given by

$$f_i(x) = b_i, \qquad i = 1, \ldots, m,$$

where $f_i \colon \mathbf{R}^n \to \mathbf{R}$ are scalar-valued functions, and $b \in \mathbf{R}^m$ is a given vector. By defining $f \colon \mathbf{R}^n \to \mathbf{R}^m$ as $f(x) = (f_1(x), \ldots, f_m(x))$, we can write the equations more compactly as

$$f(x) = b. \tag{4.1}$$

Many machine learning problems can be expressed as the problem of trying to find a solution to a system of linear or nonlinear equations in this form, where the function f is determined by the model structure with parameters x, and the vector b is the observed data, *i.e.*, the target model output.

For most cases in practice, however, we cannot find an x that satisfies all equations in (4.1) simultaneously, and we have to resort to *approximation*, *i.e.*, to find an x that approximately satisfies the equations. To quantify how well some $x \in \mathbf{R}^n$ satisfy the equations, we can define the *residual* $r \in \mathbf{R}^m$ for this x as

$$r = f(x) - b,$$

which measures how far we are from satisfying the equations exactly. The goal of approximation is to find an x that leads to the 'smallest' residual r, whose size can be measured in various ways, depends on the application scenario (as we will see in the next sections).

Example 4.1 *Linear models.* A linear model corresponds to the case where the function f in (4.1) is linear, *i.e.*, has the form

$$f(x) = Ax,$$

for some given matrix $A \in \mathbf{R}^{m \times n}$. The matrix A is often referred to as the *feature matrix*, where each row of A corresponds to the features of one data point. In this case, the system of equations we would like to solve becomes

$$Ax = b, \tag{4.2}$$

and the residual for some $x \in \mathbf{R}^n$ is given by

$$r = Ax - b.$$

The linear system (4.2) is solvable, *i.e.*, there exists some $x \in \mathbf{R}^n$ such that $r = 0$, only when $b \in \mathcal{R}(A)$, *i.e.*, the vector b can be expressed as a linear combination of the columns of A.

Example 4.2 *Matrix factorization.* In matrix factorization problems, we are given a matrix $B \in \mathbf{R}^{m \times n}$, and we would like to find two matrices $X \in \mathbf{R}^{m \times k}$ and $Y \in \mathbf{R}^{k \times n}$ such that

$$XY = B. \tag{4.3}$$

To formulate this as a system of equations in the form of (4.1), we can let the variable $x = (X, Y)$, and define the function f as

$$f(X, Y) = XY,$$

and the residual $R \in \mathbf{R}^{m \times n}$ is given by

$$R = XY - B.$$

Note that here the function f is bilinear in the variables X and Y.

The solvability of the matrix equation depends on the choice of the inner dimension k. In particular, the system (4.3) is solvable only when $k \geq \mathbf{rank}\, B$.

If either the matrix X or the matrix Y is fixed, then the system of equations (4.3) becomes a linear system of equations in the other variable. In particular, it can be expressed in the form of (4.2) by vectorizing the variable matrix and the residual matrix, and then defining the feature matrix accordingly.

4.1.2 Norm approximation

When the size of some residual $r \in \mathbf{R}^m$ is measured using a norm $\|\cdot\|$, we have the *norm approximation problem* of the form

$$\text{minimize} \quad \|r\| = \|f(x) - b\| \tag{4.4}$$

with variable $x \in \mathbf{R}^n$. Obviously, the optimal value $r^\star$ of the problem (4.4) satisfies $r^\star = 0$ if and only if the system of equations (4.1) is solvable, and in this case, any x that solves (4.4) also solves (4.1). When the optimal value $r^\star > 0$, a solution of the norm approximation problem (4.4) is sometimes called an *approximate solution* of the system of equations $f(x) = b$, in the norm $\|\cdot\|$.

Linear norm approximation

When the function f in (4.4) is linear, *i.e.*, has the form $f(x) = Ax$ with some given matrix $A \in \mathbf{R}^{m\times n}$, we have the *linear norm approximation problem*, given by

$$\text{minimize} \quad \|Ax - b\| \tag{4.5}$$

where $x \in \mathbf{R}^n$ is the variable and $A \in \mathbf{R}^{m\times n}$, $b \in \mathbf{R}^m$ are problem data.

The linear norm approximation problem (4.5) is a convex optimization problem for any choice of the norm $\|\cdot\|$, and hence can be solved efficiently. In particular, there is always at least one solution.

If $b \in \mathcal{R}(A)$, then the optimal value of the problem is zero. It is more interesting and useful in practice when $b \notin \mathcal{R}(A)$, in which case the optimal value is strictly positive, and by solving (4.5), we are looking for an approximate solution of the linear system of equations $Ax = b$, in the norm $\|\cdot\|$.

We can assume without loss of generality that the matrix A has full rank with $\mathbf{rank}\, A = n$, *i.e.*, the columns of A are linearly independent, and hence $m \geq n$. When $m = n$, the solution of (4.5) is simply $A^{-1}b$, so we can further assume that $m > n$.

Remark 4.1 The linear norm approximation problem (4.5) can be interpreted from various perspectives:

Approximation interpretation. Let $a_1, \ldots, a_n \in \mathbf{R}^m$ denote the columns of the matrix A, then for some $x \in \mathbf{R}^n$, the vector Ax can be expressed as a linear combination of these columns, *i.e.*,

$$Ax = x_1a_1 + x_2a_2 + \cdots + x_na_n.$$

Thus, by solving the problem (4.5), we are looking for a linear combination of the columns of A that best approximates the vector b, with deviation measured by the norm $\|\cdot\|$. In this context, the linear norm approximation problem is also called a *regression problem*, and the vectors $a_1, \ldots, a_n$ are called the *regressors*.

Model fitting interpretation. Let $\tilde{a}_1^T, \ldots, \tilde{a}_m^T$ denote the rows of A, then the residual vector $r = Ax - b$ can be expressed componentwise as

$$r_i = \tilde{a}_i^T x - b_i, \qquad i = 1, \ldots, m.$$

We can interpret each row $\tilde{a}_i^T$ as the *feature vector* of the ith data point, and the linear function $\tilde{a}_i^T x$ as the model output for this data point, with model parameter x. The residual r_i then measures the deviation of the model output from the observed response b_i for the ith data point, *i.e.*, the model prediction error. Hence, by solving the problem (4.5), we are looking for a model parameter x that leads to the smallest cumulative prediction error for all data points, measured by the norm $\|\cdot\|$.

Geometric interpretation. Notice that the range of A is defined as

$$\mathcal{R}(A) = \{Ax \mid x \in \mathbf{R}^n\},$$

so the linear norm approximation problem (4.5) is equivalent to

$$\begin{array}{ll} \text{minimize} & \|z - b\| \\ \text{subject to} & z \in \mathcal{R}(A) \end{array}$$

with variable $z \in \mathbf{R}^m$. This problem can be interpreted as finding the point in the subspace $\mathcal{R}(A)$ that is closest to the point b with distance measured by the norm $\|\cdot\|$, *i.e.*, finding the *projection* of the point b onto the subspace $\mathcal{R}(A)$, in the norm $\|\cdot\|$.

Estimation interpretation. Assume that the response $b \in \mathbf{R}^m$ is generated from a (noisy) *linear measurement model* with some parameter $x \in \mathbf{R}^n$ (to be estimated), given by

$$b = Ax + r,$$

where the vector $r \in \mathbf{R}^m$ is some noise or measurement error that is unknown. Assuming that smaller values of the noise r (measured by the norm $\|\cdot\|$) is more plausible than larger values, by solving the problem (4.5), we are looking for the most plausible estimation of the parameter x (which results in the smallest measurement noise). (Note that $-r$ should have been used in the linear measurement model so that the notation is consistent with the previous definition of the residual, but negating the sign here does not affect the interpretation since we have $\|u\| = \|-u\|$ for any norm $\|\cdot\|$ and $u \in \mathbf{R}^m$.) This interpretation will be discussed more formally in §4.2.1.

Example 4.3 *Least squares approximation.* A special case of the linear norm approximation problem (4.5) is when the norm $\|\cdot\|$ is chosen as the Euclidean norm or ℓ_2-norm, which is defined for any vector $u \in \mathbf{R}^n$ as

$$\|u\|_2 = (u^T u)^{1/2} = (u_1^2 + \cdots + u_n^2)^{1/2}.$$

By choosing this norm and squaring the objective, we can equivalently write the problem (4.5) as

$$\text{minimize} \quad \|Ax - b\|_2^2 = r_1^2 + \cdots + r_m^2, \tag{4.6}$$

where the objective is now the sum of squared residuals. The problem (4.6) is called the (linear) *least squares approximation problem*.

To solve the problem, we can rewrite the objective as

$$\|Ax - b\|_2^2 = x^T A^T Ax - 2b^T Ax + b^T b,$$

which is a convex quadratic function in x. Hence, the solution can be found by setting the gradient of the objective to zero:

$$\nabla \|Ax - b\|_2^2 = 2A^T Ax - 2A^T b = 0,$$

which gives the so-called *normal equations*

$$A^T Ax = A^T b.$$

The normal equations is always solvable and since we have assumed that the matrix A has full rank with $\mathbf{rank}\, A = n$, the least squares problem (4.6) has the unique analytical solution given by

$$x^\star = (A^T A)^{-1} A^T b.$$

Geometrically, least squares approximation corresponds to finding the point $Ax^\star$ in the subspace $\mathcal{R}(A)$ that is closest to the point b in terms of Euclidean distance, *i.e.*, finding the Euclidean projection of the point b onto the subspace $\mathcal{R}(A)$. Hence, the optimal point $x^\star$ must satisfy that the residual $r^\star = Ax^\star - b$ is orthogonal to the subspace $\mathcal{R}(A)$, *i.e.*,

$$(Ax^\star - b)^T Az = 0$$

for all $z \in \mathbf{R}^m$, which is equivalent to the normal equations. This idea is illustrated in figure 4.1.

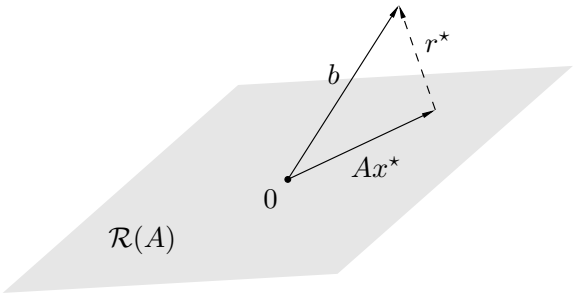

Figure 4.1 Geometric interpretation of least squares approximation. The point $Ax^\star \in \mathcal{R}(A)$ is the projection of the point b onto the subspace $\mathcal{R}(A)$, and the residual $r^\star = Ax^\star - b$ is orthogonal to the subspace.

Least absolute residuals approximation

When ℓ_1-norm is used in the norm approximation problem (4.4), we have the ℓ_1-*norm* or *least absolute residuals* approximation problem, given by

$$\text{minimize} \quad \|f(x) - b\|_1 = |r_1| + |r_2| + \cdots + |r_m|, \tag{4.7}$$

where the objective is the sum of absolute residuals.

If the function f is linear, the problem (4.7) becomes

$$\text{minimize} \quad \|Ax - b\|_1 = \textstyle\sum_{i=1}^m |a_i^T x - b_i|,$$

where a_i^T are the rows of the matrix $A \in \mathbf{R}^{m\times n}$, and can be cast as an equivalent linear program by introducing an auxiliary variable $t \in \mathbf{R}^m$, expressed as

$$\begin{array}{ll} \text{minimize} & \mathbf{1}^T t = t_1 + t_2 + \cdots + t_m \\ \text{subject to} & -t \preceq Ax - b \preceq t, \end{array}$$

with variables $x \in \mathbf{R}^n$ and $t \in \mathbf{R}^m$.

Least square residuals approximation

When ℓ_2-norm is used in the norm approximation problem (4.4), it is common in practice to square the objective, which leads to an equivalent formulation, given by

$$\text{minimize} \quad \|f(x) - b\|_2^2 = r_1^2 + r_2^2 + \cdots + r_m^2, \tag{4.8}$$

where the objective is the sum of squared residuals, and is therefore called the *least square residuals* (or simply *least squares*) approximation problem. (This problem is sometimes also called the (squared) ℓ_2*-norm approximation problem*.)

If the function f is linear, the problem (4.8) becomes the least squares approximation problem (4.6), where an analytical solution exists (as discussed in example 4.3).

Least maximum residuals approximation

When ℓ_∞-norm is used in the norm approximation problem (4.4), we have the *ℓ_∞-norm* or *least maximum residuals* approximation problem, given by

$$\text{minimize} \quad \|f(x) - b\|_\infty = \max\{|r_1|, \ldots, |r_m|\}, \tag{4.9}$$

where the objective is the maximum of the absolute residual. The problem (4.9) is also known as the *minimax approximation problem*, or *Chebyshev approximation problem*.

If the function f is linear, the problem (4.9) becomes

$$\text{minimize} \quad \|Ax - b\|_\infty = \max_{i=1,\ldots,m} |a_i^T x - b_i|,$$

and similar to the least absolute residuals approximation problem (4.7), it can be cast as an equivalent linear program by introducing an auxiliary variable $t \in \mathbf{R}$, given by

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & -t\mathbf{1} \preceq Ax - b \preceq t\mathbf{1}, \end{array}$$

with variables $x \in \mathbf{R}^n$ and $t \in \mathbf{R}$.

Weighted norm approximation

Let $W \in \mathbf{R}^{m\times m}$ be a nonsingular matrix, and suppose $\|\cdot\|$ is a norm on $\mathbf{R}^m$. It can be shown that the function defined as

$$\|u\|_W = \|Wu\|$$

for all $u \in \mathbf{R}^m$ is also a norm on $\mathbf{R}^m$, and is called a *weighted norm*. By using a weighted norm in the norm approximation problem (4.4), we have the *weighted norm approximation problem*, given by

$$\text{minimize} \quad \|f(x) - b\|_W = \|W(f(x) - b)\| \tag{4.10}$$

with variable $x \in \mathbf{R}^n$. The matrix W is sometimes called the *weighting matrix* and is often chosen to be diagonal, so that it adds different weights to each component of the residual $r = f(x) - b$. When $W = I$ is the identity matrix, the problem (4.10) reduces to the standard norm approximation problem (4.4).

If the function f is linear with $f(x) = Ax$, the weighted norm approximation problem (4.10) can be interpreted as first transforming the problem data A and b using the weighting matrix W, *i.e.*, let $\tilde{A} = WA$ and $\tilde{b} = Wb$, and then solving the standard norm approximation problem

$$\text{minimize} \quad \|\tilde{A}x - \tilde{b}\|$$

with variable $x \in \mathbf{R}^n$.

Matrix norm approximation

The idea of norm approximation can be extended to the case where the residual is given by a matrix. As a basic example, consider the matrix factorization equation (4.3) with data $B \in \mathbf{R}^{m\times n}$, where the residual for some $X \in \mathbf{R}^{m\times k}$ and $Y \in \mathbf{R}^{k\times n}$ is given by the matrix

$$R = XY - B.$$

Suppose, for example, that we want to measure the size of the residual R using the *Frobenius norm*, which is defined for any matrix $U \in \mathbf{R}^{m\times n}$ as

$$\|U\|_F = (\mathbf{tr}(U^TU))^{1/2} = \left(\sum_{i=1}^{m}\sum_{j=1}^{n} U_{ij}^2\right)^{1/2},$$

then (by squaring the objective) we have the *least squares matrix factorization* problem, given by

$$\text{minimize} \quad \|R\|_F^2 = \|XY - B\|_F^2 \tag{4.11}$$

with variables $X \in \mathbf{R}^{m\times k}$ and $Y \in \mathbf{R}^{k\times n}$, where the objective is the sum of the squares of all entries in the residual matrix R. The problem (4.11) is also known as the problem of *principal component analysis*, since when $k < \mathbf{rank}\, B$, a solution of the problem gives the best rank-k approximation of the matrix B, in terms of the least squares deviation.

Notice that the Frobenius norm of some matrix can be interpreted as the Euclidean norm of the vector from vectorizing the matrix, according to some rule, *e.g.*, by stacking the columns vertically. Hence, the problem (4.11) can be expressed as a standard least squares approximation problem (4.8) by transforming residual matrix $R \in \mathbf{R}^{m\times n}$ into a vector $r \in \mathbf{R}^{mn}$, and then define the objective as the size of the resulting residual vector measured by the (squared) Euclidean norm.

The problem (4.11) is not convex (indeed, since the residual is a bilinear function of X and Y), but analytical solution exists via the *singular value decomposition* (see §1.2.2). Although analytical solutions do not generally exist when other matrix norms are used instead of the Frobenius norm in the problem (4.11), one can often find a good approximate solution via alternate convex optimization (see §3.1.3).

4.1.3 Penalty function approximation

The idea of norm approximation can be generalized by measuring the size of the residual via some *penalty functions*. Specifically, suppose we want to (possibly approximately) solve the system of equations $f_i(x) = b_i$, $i = 1, \ldots, m$, and let $\phi\colon \mathbf{R} \to \mathbf{R}$ be a scalar-valued penalty function, then the size of the residual $r \in \mathbf{R}^m$, where, recalling that, $r_i = f_i(x) - b_i$, is measured as the total penalty

$$\phi(r_1) + \cdots + \phi(r_m),$$

i.e., the sum of the penalties assessed for each component of the residual. Different x leads to different residual r, and hence different total penalties. The approximation problem with such a size measure is then written as

$$\text{minimize} \quad \phi(r_1) + \cdots + \phi(r_m) = \textstyle\sum_{i=1}^{m} \phi(f_i(x) - b_i) \tag{4.12}$$

with variable $x \in \mathbf{R}^n$, and is called the *penalty function approximation problem*. The problem (4.12) is sometimes written in a constrained form as

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^m \phi(r_i) \\ \text{subject to} & r = f(x) - b, \end{array} \tag{4.13}$$

where the variables are $x \in \mathbf{R}^n$ and $r \in \mathbf{R}^m$, and the function $f\colon \mathbf{R}^n \to \mathbf{R}^m$ is given by $f(x) = (f_1(x), \ldots, f_m(x))$. Obviously, the problems (4.12) and (4.13) are equivalent, by eliminating the auxiliary variable r in the constrained form.

In the most general case, the function ϕ is chosen to be nonnegative, and to satisfy $\phi(0) = 0$. In many cases, it is also chosen to be symmetric. When the function f is linear, and if ϕ is convex, then the problem (4.12) is a convex optimization problem (since a convex function composed with any affine function is convex). However, none of these properties about the penalty function is strictly necessary and we will not require them in the following analysis.

Penalty function interpretation of norms

Many norm approximation problems presented above can be interpreted as special cases of penalty function approximation. As a basic example, the norm approximation problem with ℓ_1-norm, *i.e.*, the least absolute residuals problem (4.7), can be interpreted as a penalty function approximation problem with the *absolute value penalty function*, given by

$$\phi(u) = |u|,$$

for all $u \in \mathbf{R}$. As another example, the least square residuals problem (4.8) can be interpreted as a penalty function approximation problem with the *quadratic penalty function* (or *least squares penalty function*), given by

$$\phi(u) = u^2.$$

This idea can be generalized to any *ℓ_p-norm approximation problem* with $p \in [1, \infty)$, where the ℓ_p-norm objective for residual $r \in \mathbf{R}^m$ is given by

$$\|r\|_p = (|r_1|^p + |r_2|^p + \cdots + |r_m|^p)^{1/p}.$$

Similar to the ℓ_2-norm case, by raising the objective to the power of p, we can equivalently write the ℓ_p-norm approximation problem as

$$\text{minimize} \quad \|f(x) - b\|_p^p = |r_1|^p + |r_2|^p + \cdots + |r_m|^p,$$

which can be interpreted as a penalty function approximation problem with the absolute value penalty function raised to the power of p, defined as

$$\phi(u) = |u|^p$$

for any $u \in \mathbf{R}$.

As a more complex example, consider the norm approximation problem with the ℓ_∞-norm, which corresponds to the least maximum residuals problem (4.9).

This norm approximation problem can be interpreted as a penalty function approximation problem with the *Chebyshev penalty function*, which is defined for some residual $r \in \mathbf{R}^m$ as

$$\phi(r_i) = \begin{cases} |r_i|, & |r_i| = \max\{|r_1|, \ldots, |r_m|\} \\ 0, & \text{otherwise.} \end{cases} \tag{4.14}$$

Note that unlike the previous examples, the Chebyshev penalty function is not applied to each entry of the residual vector independently, but instead depends on all components of the residual.

Remark 4.2 It is often seen in the literature that the absolute value penalty $\phi(u) = |u|$, quadratic penalty $\phi(u) = u^2$, and Chebyshev penalty given by (4.14) are called the *ℓ_1-norm penalty*, *ℓ_2-norm penalty*, and *ℓ_∞-norm penalty* functions, respectively, since the corresponding penalty function approximation problem is equivalent to the respective ℓ_p-norm approximation problems with $p = 1$, 2, and ∞. (The quadratic penalty is called the *least squares penalty* for the same reason.) In the most general case, the penalty function $\phi(u) = |u|^p$ for $p \in [1, \infty)$ is often called the *ℓ_p-norm penalty*. However, we should be careful about these names, since these penalty functions are not actually the ℓ_p-norms by themselves.

Some other penalty functions

There are many parameterized penalty functions that are very useful in practice, which can be roughly considered as some kind of variations of the penalty functions based on vector norms as discussed previously.

The *deadzone-linear* penalty function with parameter $\alpha \geq 0$ is defined as

$$\phi(u) = \begin{cases} |u| - \alpha, & |u| > \alpha \\ 0, & \text{otherwise,} \end{cases} \tag{4.15}$$

for any $u \in \mathbf{R}$. This penalty function does not penalize small residuals with absolute value less than or equal to α, and penalizes large residuals linearly. When $\alpha = 0$, the deadzone-linear penalty function reduces to the absolute value penalty function $\phi(u) = |u|$.

The *log-barrier* penalty function with parameter $\alpha > 0$ is defined as

$$\phi(u) = \begin{cases} -\alpha^2 \log(1 - (u/\alpha)^2), & |u| < \alpha \\ \infty, & \text{otherwise,} \end{cases} \tag{4.16}$$

for any $u \in \mathbf{R}$. The log-barrier penalty is very close to the quadratic function $\phi(u) = u^2$ for small residuals, but penalizes large residuals more heavily, and becomes infinite when the absolute residual is larger than the threshold α.

The deadzone-linear and log-barrier penalty functions are shown in figure 4.2 with comparison to the quadratic penalty. Notice that the log-barrier penalty is very close to the quadratic function for small residuals.

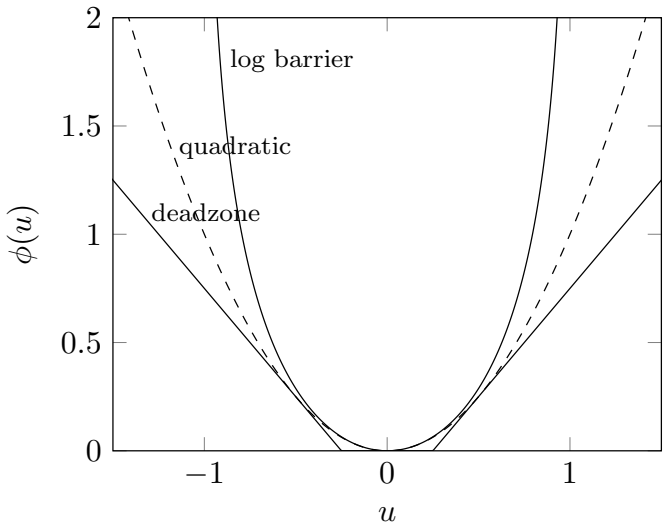

Figure 4.2 Graph of the quadratic penalty $\phi(u) = u^2$ (shown dashed), the deadzone-linear penalty with parameter $\alpha = 0.25$, and the log-barrier penalty with parameter $\alpha = 1$.

Selection of penalty functions

It is easy to see that for any fixed penalty function, scaling it by some positive constant does not change the solution of the resulting penalty function approximation problem (4.12), since this is equivalent to scaling the objective by the same constant. However, we might want to carefully select the *shape* of a penalty function $\phi\colon \mathbf{R} \to \mathbf{R}$ for different applications, which plays an important role in determining the solution characteristics.

Roughly speaking, penalty function ϕ represents our *irritation* towards residuals of different sizes. If $\phi(u)$ is very small (or even zero) for small u, then we care very little (or not at all) by small residuals, and hence would like to accept them. If $\phi(u)$ grows quickly as u increases, then we are very irritated by large residuals, and would like to avoid them as much as possible. If $\phi(u)$ becomes even infinite for some u, *e.g.*, outside some interval as in (4.16), it means that these residuals are completely unacceptable. This simple interpretation gives insight into the solution characteristics of a penalty function approximation problem, as well as guidelines for choosing or designing a penalty function.

As a basic example, consider the penalty functions $\phi_1(u) = |u|$ and $\phi_2(u) = u^2$, corresponding to the ℓ_1-norm and the (squared) ℓ_2-norm approximation problems, respectively. For $u = 1$, both penalty functions give the same penalty value of 1. For small u values, we have $\phi_1(u) \gg \phi_2(u)$, so the absolute value penalty function penalizes small residuals more heavily than the quadratic penalty function, and hence tends to reduce small residuals to zero more effectively. For large u values, however, we have $\phi_1(u) \ll \phi_2(u)$, so the quadratic penalty function puts much more emphasis on reducing large residuals. Put together, we can expect that solution

of the ℓ_1-norm approximation problems tends to have lots of zero residuals, but might also have some residuals with large absolute values, compared to the ℓ_2-norm approximation solution. On the other hand, solution of the ℓ_2-norm approximation problems tends to have fewer large residuals compared to the ℓ_1-norm approximation solution, but most residuals might be nonzero.

Numerical example

We compare the solution characteristics between various penalty functions via a simple numerical example, to further illustrate the discussion above. Consider the linear approximation problem with data $A \in \mathbf{R}^{100\times 30}$ and $b \in \mathbf{R}^{100}$, where the entries of A and b are chosen randomly. We compare the optimal residual $r^\star = Ax^\star - b$ obtained by solving five different penalty function approximation problems, with the following penalty functions:

- Absolute value penalty $\phi(u) = |u|$, corresponding to the ℓ_1-norm approximation problem.
- Quadratic penalty $\phi(u) = u^2$, corresponding to the least squares approximation problem.
- Chebyshev penalty given by (4.14), corresponding to the ℓ_∞-norm approximation problem.
- Deadzone-linear penalty (4.15) with parameter $\alpha = 0.5$.
- log-barrier penalty (4.16) with parameter $\alpha = 1$.

Note that the first two penalty functions are special cases of the penalty function $\phi(u) = |u|^p$ with $p = 1$ and $p = 2$, respectively. Figure 4.3 shows the histogram of the optimal residual amplitudes obtained from the five different penalty function approximation problems, shown in the order listed above from top to bottom.

From the graphs of the penalty functions, we notice the following characteristics:

- The absolute value penalty penalizes small residuals most heavily, but only puts relatively small penalty for large residuals.
- The quadratic penalty penalizes only puts very small penalty for small residuals, but puts strong penalty for large residuals.
- The deadzone-linear penalty function does not penalize small residuals with absolute value less than 0.5, and penalizes large residuals very much like the absolute value penalty.
- The log-barrier penalty function penalizes small residuals very similarly to the quadratic penalty, but the penalty grows very quickly as the residual increases, and becomes infinite for residuals with absolute value larger than 1.

For the Chebyshev penalty, it only penalizes the largest absolute residual, in the same way as the absolute value penalty, and all other residuals are not active in the total penalty calculation.

These properties are reflected in the distribution of the optimal residual amplitudes shown in figure 4.3:

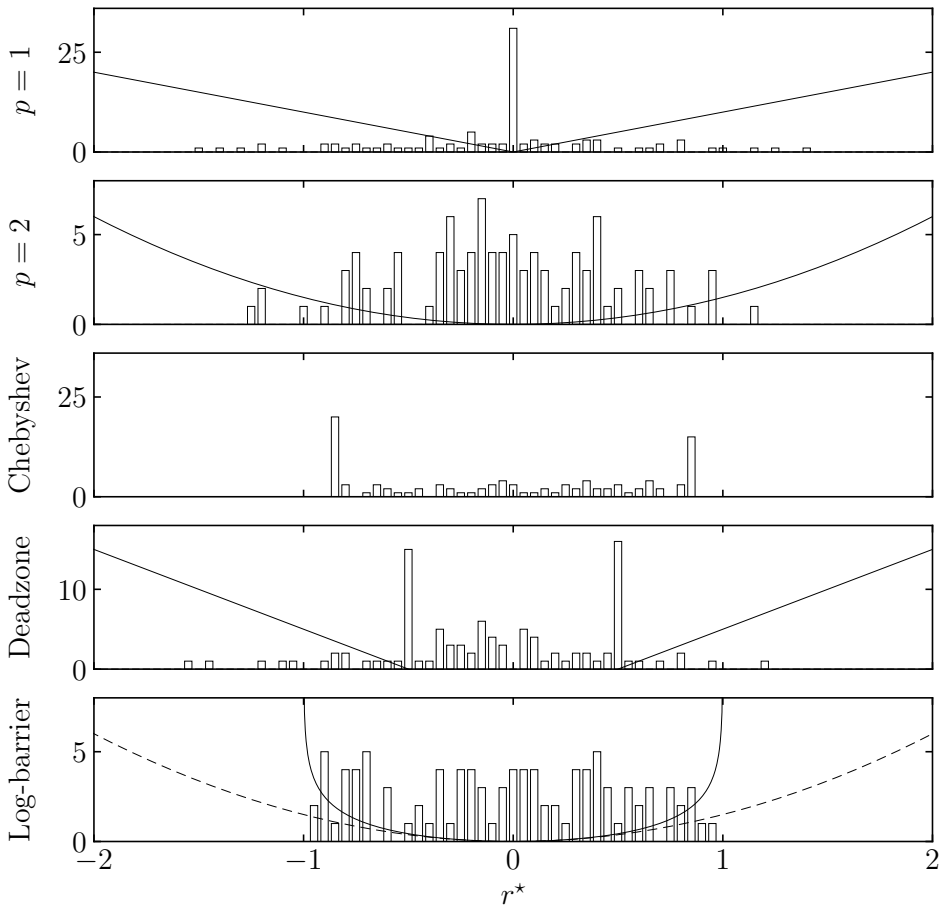

Figure 4.3 Histogram of the optimal residual amplitudes from five different penalty function approximation problems. The (scaled) penalty functions are shown as solid lines (except for the Chebyshev penalty) in the corresponding plots for reference. The quadratic penalty function is also shown dashed in the log-barrier plot.

- Most residuals obtained from the ℓ_1-norm approximation are either zero or very close to zero, but there are also many large residuals.
- The quadratic penalty function approximation solution has relatively few large residuals, but most residuals have moderate magnitude.
- For the Chebyshev penalty function approximation solution, most of the residual amplitudes are concentrated at the maximum absolute residual amplitude value, *i.e.*, at the two ends of the residual range, where all other residuals are bounded in between.
- The residuals obtained with the deadzone-linear penalty function aggregate at the two ends of the deadzone interval, *i.e.*, at ± 0.5, and most residual amplitudes are within the deadzone, where no penalty is assessed. This result is similar to the Chebyshev penalty approximation solution, except that there still exist some residual amplitudes outside the deadzone interval.
- The log-barrier penalty function results in residuals that are all within the threshold of 1, but otherwise similar to the quadratic penalty function solution, where most residuals have moderate magnitude.

Remark 4.3 *Sparse approximation.* We have seen in the previous example that the ℓ_1-norm approximation solution tends to have many zero residual amplitudes. In other words, the optimal residual vector obtained from the ℓ_1-norm approximation problem is often *sparse*, *i.e.*, many of its entries are zero, which indicates that many of the equations $f_i(x) = b_i$ are satisfied exactly by the solution. Hence, the ℓ_1-norm approximation problem is sometimes called the *sparse approximation problem*, and the ℓ_1-norm (or the absolute value penalty) is said to *induce sparsity*.

4.1.4 Synthetic penalty functions

According to the previous idea, we can design various penalty functions with different shapes to achieve different solution characteristics. Here we present some basic examples.

Quantile penalty function

The *quantile penalty function* with parameter $\alpha \in (0, 1)$ is defined as

$$\begin{aligned} \phi(u) &= \max\{\alpha u, (\alpha - 1)u\} \\ &= \begin{cases} \alpha|u|, & u \geq 0 \\ (1-\alpha)|u|, & \text{otherwise}, \end{cases} \end{aligned} \tag{4.17}$$

for any $u \in \mathbf{R}$. The quantile penalty function can be considered as a 'tilted' absolute value penalty, since it adds linear penalty to the residual, but with different scales for the positive and negative components. When $\alpha = 0.5$, the quantile penalty function reduces to the absolute value penalty function $\phi(u) = |u|$ (scaled by $1/2$). Figure 4.4 shows the graph of the quantile penalty function with different α values.

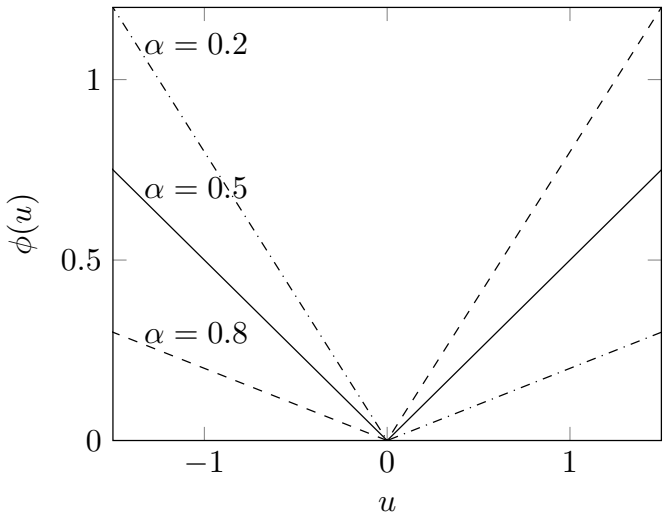

Figure 4.4 Graph of the quantile penalty function with parameters $\alpha = 0.5$ (shown solid), $\alpha = 0.2$ (shown dashdotted), and $\alpha = 0.8$ (shown dashed).

We can evaluate the solution characteristics of this penalty function. For example, consider the quantile penalty function (4.17) with $\alpha = 0.5$ (*i.e.*, the absolute value penalty function) and $\alpha = 0.8$. According to figure 4.4, when $\alpha = 0.5$, the penalty function penalizes residuals with the same absolute value equally, so the solution tends to balance the number of positive and negative residuals. In other words, it balances the number of *overestimation* and *underestimation*. When $\alpha = 0.8$, the penalty function penalizes positive residuals more heavily than negative residuals (given that they have the same absolute value), so the solution tends to have more negative residuals than positive ones, *i.e.*, tends to underestimate more often than overestimate.

Example 4.4 *Quantile regression.* Given the data $A \in \mathbf{R}^{m\times n}$ and $b \in \mathbf{R}^m$, the linear approximation problem

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^m \phi(r_i) \\ \text{subject to} & r = Ax - b, \end{array}$$

with variable $x \in \mathbf{R}^n$ and $r \in \mathbf{R}^m$, where ϕ is given by the quantile penalty function (4.17), is sometimes called the *quantile regression*. The name follows from the fact that the optimal residual of the problem satisfies that approximately α-fraction of the entries are negative, and $(1-\alpha)$-fraction of the entries are positive, *i.e.*, the α-quantile of the residual amplitudes distribution is zero. This aligns well with our expectation about the solution characteristics of the quantile penalty function.

We illustrate this idea via a simple numerical example. Consider the linear approximation problem with data $A \in \mathbf{R}^{500\times 30}$ and $b \in \mathbf{R}^{500}$, where the entries of A and b are chosen randomly. We solve the quantile regression problem with $\alpha = 0.5$, 0.8, and 0.2, respectively. Figure 4.5 shows the histogram of the optimal residual amplitudes. When $\alpha = 0.5$, the residual amplitudes are roughly symmetrically distributed around

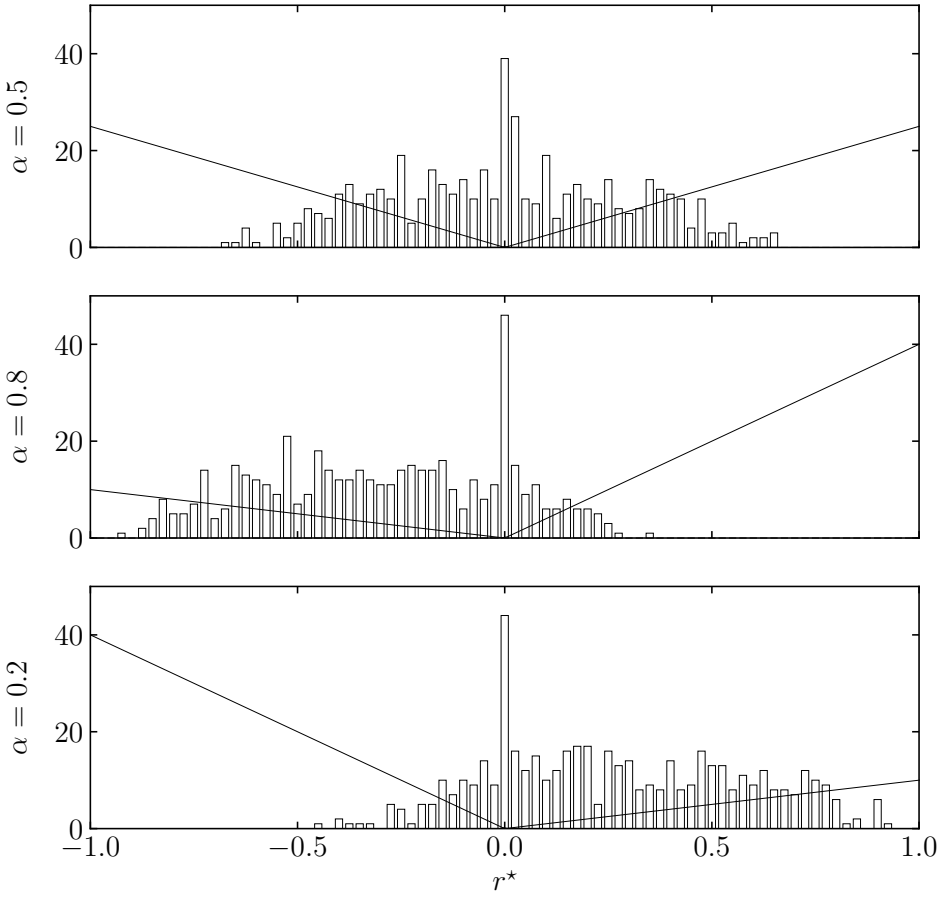

Figure 4.5 Histogram of the optimal residual amplitudes from the quantile regression problem with parameters $\alpha = 0.5$, 0.8, and 0.2. The (scaled) quantile penalty functions are shown as solid lines in the corresponding plots for reference.

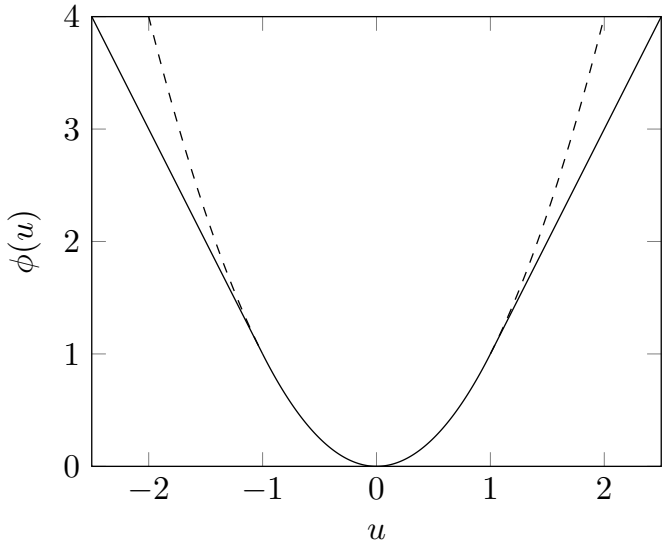

Figure 4.6 Graph of the Huber penalty function with parameter $M = 1$, shown in solid line. The dashed lines show the extended quadratic part of the penalty function, towards the linear region.

zero, which is similar to the results seen in the ℓ_1-norm approximation solution (figure 4.3, top). For the other two cases, the residual amplitudes distribution skews to the negative side when $\alpha = 0.8$, and to the positive side when $\alpha = 0.2$. Specifically, when $\alpha = 0.8$, approximately 80% of the residuals are negative, and when $\alpha = 0.2$, approximately 20% of the residuals are negative, while the majority of the residuals are still zero in both cases.

Huber penalty function

The *Huber penalty function* (or *robust least squares penalty*) with parameter $M > 0$ is defined as

$$\phi(u) = \begin{cases} u^2, & |u| \leq M \\ M(2|u| - M), & \text{otherwise,} \end{cases} \tag{4.18}$$

for any $u \in \mathbf{R}$, which penalizes small residuals quadratically as in the least square residuals problem (4.8), but switches to a linear penalty for large residuals. This penalty function hence combines the properties from both the quadratic penalty and the absolute value penalty. Figure 4.6 shows the graph of the Huber penalty function with parameter $M = 1$.

The Huber penalty function is often used in approximation problems with *outliers* in the data, whose residual values $f_i(x) - b_i$ are significantly larger than the others for any x values. Approximation problems with quadratic penalty function tend to be heavily influenced by outliers, since the quadratic penalty increases very quickly as the residual values grow larger. By using the Huber penalty function

instead, we retain the behavior for small residuals as in the quadratic penalty, while the influence of the outliers can be significantly reduced, since the penalty now grows only linearly for large residuals.

Example 4.5 *Robust regression.* Given the data $A \in \mathbf{R}^{m\times n}$ and $b \in \mathbf{R}^m$, the linear approximation problem

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^m \phi(r_i) \\ \text{subject to} & r = Ax - b, \end{array}$$

with variable $x \in \mathbf{R}^n$ and $r \in \mathbf{R}^m$, where ϕ is given by the Huber penalty function (4.18), is sometimes called the *robust regression*. This name follows from the property that it is less sensitive to outliers in the data compared to the standard least squares problem.

The following numerical example illustrates this idea. Suppose we are given a dataset of $m = 110$ points (t_i, y_i), $i = 1, \ldots, m$, on a 2-dimensional plane, shown as circles in figure 4.7. It is easily seen that this dataset can be affinely approximated as $y_i \approx x_1 t_i + x_2$, except for several outliers that are far away from the main cluster of points, at the upper left and lower right corners of the plot. To find such an affine approximation, we consider the problem

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^m \phi(r_i) \\ \text{subject to} & r_i = x_1 t_i + x_2 - y_i, \quad i = 1, \ldots, m, \end{array}$$

with variable $x \in \mathbf{R}^2$ and $r \in \mathbf{R}^m$, under both the least squares penalty $\phi(u) = u^2$ and the Huber penalty (4.18) with parameter $M = 1$. The dashed line in figure 4.7 shows the least squares approximation result, which is heavily influenced by the outliers. The solid line shows the robust regression result using the Huber penalty function, which fits the main cluster of points much better by reducing the penalty values assigned to the outliers.

Reverse Huber penalty function

We can also swap the roles of the quadratic and linear parts in the Huber penalty function (4.18) to obtain the *reverse Huber penalty function* (or the *BerHu penalty*), which is defined as

$$\phi(u) = \begin{cases} |u|, & |u| \leq M \\ u^2/(2M) + M/2, & \text{otherwise}, \end{cases}$$

for any $u \in \mathbf{R}$, where $M > 0$ is a parameter. This penalty function penalizes small residuals linearly, but switches to a quadratic penalty for large residuals. Figure 4.8 shows the graph of the reverse Huber penalty function with parameter $M = 1$.

The reverse Huber penalty function is often used in applications where we want to strongly penalize large residuals, as in the quadratic penalty function, while still retaining some sparsity in the optimal residual vector, as induced by the absolute value penalty function.

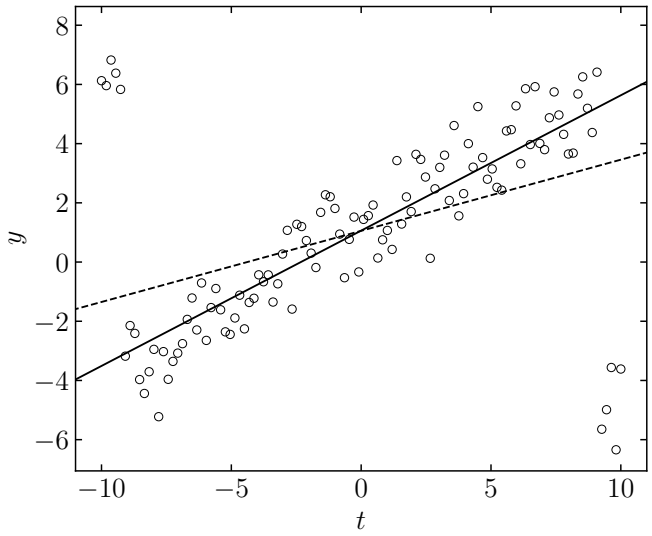

Figure 4.7 *Robust regression.* Plot of the dataset (t_i, y_i), $i = 1, \ldots, 110$, shown as circles. The dashed line shows the least squares approximation result, while the solid line shows the robust regression result using the Huber penalty function with parameter $M = 1$.

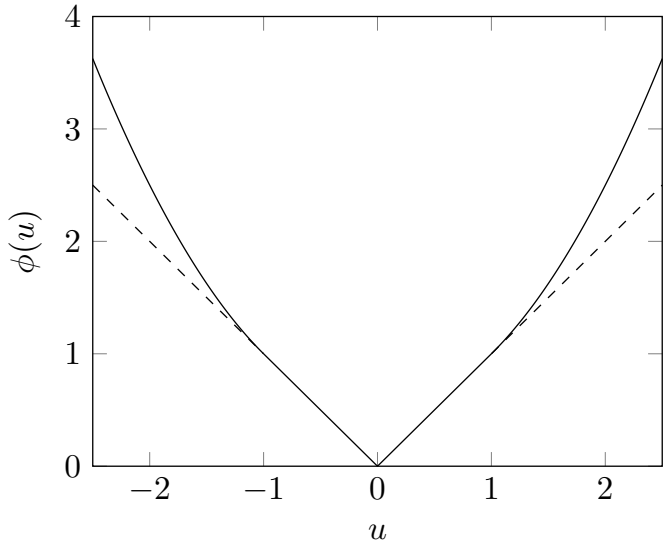

Figure 4.8 Graph of the reverse Huber penalty function with parameter $M = 1$, shown in solid line. The dashed lines show the extended linear part of the penalty function, towards the quadratic region.

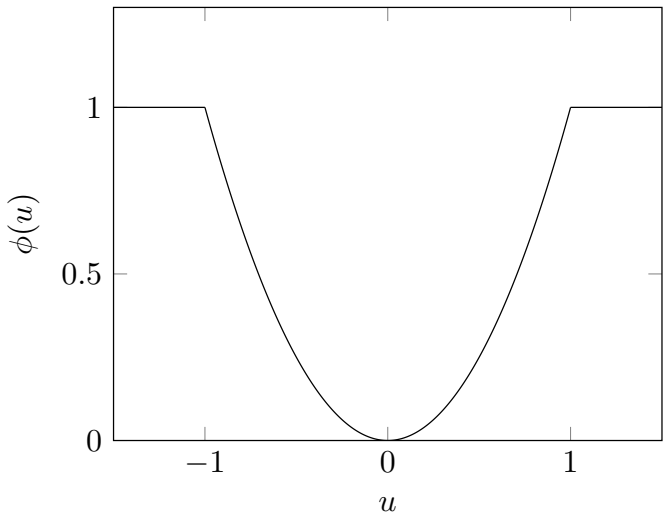

Figure 4.9 Graph of the truncated quadratic penalty function for $M = 1$.

Truncated penalty functions

Another option of dealing with outliers is to *truncate*, or *clip* a penalty function beyond some threshold. For example, a quadratic penalty function can be clipped as

$$\phi(u) = \begin{cases} u^2, & |u| \leq M \\ M^2, & \text{otherwise,} \end{cases} \tag{4.19}$$

where $M > 0$ is a parameter, and its graph is shown in figure 4.9. This idea can be readily generalized to other penalty functions as well.

The truncated quadratic penalty function (4.19) behaves exactly the same as least squares for small residuals, but assesses a constant penalty for residuals larger than M. Hence, it is expected to be even more robust to outliers compared to the Huber penalty function. Unfortunately, such a truncated penalty function is not convex, and the resulting approximation problem becomes a hard combinatorial optimization problem in general.

A special case of the truncated penalty function is when a penalty function is constant for all nonzero residuals, *i.e.*,

$$\phi(u) = \begin{cases} 0, & u = 0 \\ M, & \text{otherwise,} \end{cases}$$

where $M > 0$ is some constant. When $M = 1$, this penalty function agrees with the *cardinality function*, which, when defined on $\mathbf{R}$, returns 0 if the input is zero, and 1 otherwise, and when defined on $\mathbf{R}^m$, returns the number of nonzero entries in the input vector.

Using the cardinality penalty function in an approximation problem consists in minimizing the *number* of nonzero residuals, *i.e.*, encouraging sparsity in the optimal residual vector. We may notice that this property is similar to the absolute

value penalty function. In fact, these two penalty functions are closely related, in the sense that the latter is the *convex envelope* of the former, *i.e.*, the largest convex function that is less than or equal to the cardinality penalty function everywhere (see example 2.11, page 38). Therefore, in practice, to avoid solving a hard combinatorial optimization problem, the ℓ_1-norm approximation problem is often used as a surrogate, or heuristic, for the cardinality penalty function approximation problem, and turns out to work quite well in many applications.

4.2 Maximum likelihood estimation

Consider a family of probability distributions on $\mathbf{R}^m$, given by

$$\{p_x \colon \mathbf{R}^m \to \mathbf{R}_+ \mid x \in \mathbf{R}^n\},$$

where each distribution p_x is *parameterized* or *indexed* by $x \in \mathbf{R}^n$. Suppose the data or observation $y \in \mathbf{R}^m$ is assumed to be a random variable, then for each fixed $x \in \mathbf{R}^n$, the value $p_x(y)$ represents the probability (density) of observing y under the parameter x. Conversely, for fixed data $y \in \mathbf{R}^m$, the value $p_x(y)$ is called the *likelihood* of the parameter $x \in \mathbf{R}^n$, given the observation y. Therefore, with a slight abuse of notation, the function p_x is often called the *likelihood function* of the parameter $x \in \mathbf{R}^n$.

It is generally more convenient to work with the logarithm of the likelihood function, which is called the *log-likelihood function*, denoted as $l \colon \mathbf{R}^n \to \mathbf{R}$, with

$$l(x) = \log p_x(y),$$

for all $x \in \mathbf{R}^n$. A wide range of machine learning problems related to statistical estimation can be formulated as finding a parameter x that has the maximum likelihood (or log-likelihood) under the observed data y, which is called the *maximum likelihood estimation* (MLE) problem, given by

$$\text{maximize} \quad l(x) = \log p_x(y), \tag{4.20}$$

where, note that, the variable is $x \in \mathbf{R}^n$ and the data is $y \in \mathbf{R}^m$.

The MLE problem (4.20) is a convex optimization problem (or specifically, a concave maximization problem) if the log-likelihood function l is concave in x, for each fixed data y. It is sometimes conceptually useful to write the problem (4.20) as an equivalent minimization problem by negating the objective, as

$$\text{minimize} \quad -l(x) = -\log p_x(y),$$

where the objective $-l \colon \mathbf{R}^n \to \mathbf{R}$ is called the *negative log-likelihood function*.

4.2.1 Linear approximation

Suppose we are given a dataset (a_i, y_i), $i = 1, \ldots, m$, generated from a *linear measurement model*

$$y_i = a_i^T x + v_i, \tag{4.21}$$

where $x \in \mathbf{R}^n$ is an unknown parameter vector to be estimated, $a_i \in \mathbf{R}^n$ are known measurement vectors, $y_i \in \mathbf{R}$ are observed responses, and $v_i \in \mathbf{R}$ are the measurement noise. Assuming that the noise v_i are *independent and identically distributed* (*i.e.*, IID) for $i = 1, \ldots, m$, drawn from some distribution with probability density function $p\colon \mathbf{R} \to \mathbf{R}_+$, then the likelihood of x given the observed response $y \in \mathbf{R}^m$ for all samples is

$$p_x(y) = \prod_{i=1}^{m} p(y_i - a_i^T x),$$

so the log-likelihood function is

$$l(x) = \log p_x(y) = \sum_{i=1}^{m} \log p(y_i - a_i^T x).$$

With this function l, the MLE problem (4.20) becomes

$$\text{maximize} \quad l(x) = \textstyle\sum_{i=1}^{m} \log p(y_i - a_i^T x) \tag{4.22}$$

with variable $x \in \mathbf{R}^n$.

Gaussian noise

Suppose the noise v_i in the linear measurement model (4.21) are IID and drawn from a Gaussian distribution $\mathcal{N}(0, \sigma^2)$ with mean zero and variance σ^2, where the probability density function is given by

$$p(u) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{u^2}{2\sigma^2}\right),$$

for any $u \in \mathbf{R}$. Let $A = \begin{bmatrix} a_1 & \cdots & a_m \end{bmatrix}^T \in \mathbf{R}^{m \times n}$, the corresponding log-likelihood function is then

$$\begin{aligned} l(x) &= \sum_{i=1}^{m} \log p(y_i - a_i^T x) \\ &= \sum_{i=1}^{m} \left(-\frac{1}{2}\log(2\pi\sigma^2) - \frac{(y_i - a_i^T x)^2}{2\sigma^2}\right) \\ &= -\frac{m}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\|Ax - y\|_2^2. \end{aligned}$$

Noticing that the first term $-(m/2)\log(2\pi\sigma^2)$ is constant so can be removed from the objective without influencing the solution, and then by scaling the objective with $2\sigma^2$, the problem (4.22) reduces to

$$\text{maximize} \quad -\|Ax - y\|_2^2$$

with variable $x \in \mathbf{R}^n$, which is exactly the least squares approximation problem (4.8) with linear f.

In other words, the least squares approximation problem can be interpreted under a statistical framework as the MLE problem of a linear measurement model (4.21) with IID Gaussian noise.

Laplacian noise

If the noise v_i follow a Laplace distribution with mean zero and scale parameter $b > 0$, *i.e.*, with probability density function

$$p(u) = \frac{1}{2b} \exp\left(-\frac{|u|}{b}\right),$$

for any $u \in \mathbf{R}$, then the corresponding log-likelihood function is

$$\begin{aligned} l(x) &= \sum_{i=1}^{m} \log p(y_i - a_i^T x) \\ &= \sum_{i=1}^{m} \left(-\log(2b) - \frac{|y_i - a_i^T x|}{b}\right) \\ &= -m \log(2b) - \frac{1}{b}\|Ax - y\|_1, \end{aligned}$$

where $a_1^T, \ldots, a_m^T$ are the rows of $A \in \mathbf{R}^{m \times n}$. Similarly, by removing the constant term and scaling the objective, the MLE problem (4.22) with this log-likelihood function reduces to

$$\text{maximize} \quad -\|Ax - y\|_1$$

with variable $x \in \mathbf{R}^n$, which is the ℓ_1-norm approximation problem (4.7) when the function f is linear.

Uniform noise

When the noise v_i are from a uniform distribution over the interval $[-b, b]$ for some $b > 0$, *i.e.*, with probability density function

$$p(u) = \begin{cases} 1/(2b), & |u| \leq b \\ 0, & \text{otherwise}, \end{cases}$$

for any $u \in \mathbf{R}$. By defining $\log 0 = -\infty$, the corresponding log-likelihood function is expressed as

$$l(x) = \sum_{i=1}^{m} \log p(y_i - a_i^T x) = \begin{cases} -m \log(2b), & \|Ax - y\|_\infty \leq b \\ -\infty, & \text{otherwise}, \end{cases}$$

where $A \in \mathbf{R}^{m \times n}$ is defined as before. Hence, any $x \in \mathbf{R}^n$ that satisfies the constraint $\|Ax - y\|_\infty \leq b$ maximizes this log-likelihood, and is hence a solution to the MLE problem (4.22) with uniform noise.

MLE interpretation of penalty function approximation

In the most geneal case, by negating the objective, the MLE problem of some linear measurement model given by (4.22) can be interpreted as a penalty function

approximation problem in the form (4.12), with linear (or really, affine) residuals $r_i = a_i^T x - y_i$ and penalty function

$$\phi(u) = -\log p(u),$$

for any $u \in \mathbf{R}$. (Assuming that the density p is symmetric, *i.e.*, $p(u) = p(-u)$ for all $u \in \mathbf{R}$.)

Conversely, any linear penalty function approximation problem in the form

$$\text{minimize} \quad \sum_{i=1}^m \phi(y_i - a_i^T x)$$

with variable $x \in \mathbf{R}^n$ and penalty function $\phi\colon \mathbf{R} \to \mathbf{R}$ can be interpreted as the MLE problem (4.22) with noise density

$$p(u) = \frac{\exp(-\phi(u))}{\int \exp(-\phi(z))\, dz}. \tag{4.23}$$

With this observation, we can interpret the solution characteristics of different penalty function approximation problems from a statistical perspective, by examining the corresponding noise distributions. For example, if some penalty function ϕ penalizes large residuals heavily, then the corresponding noise distribution (4.23) has very small tails, *i.e.*, large noise values are very unlikely to occur. Therefore, solution of the MLE problem (4.22) under this noise distribution tends to avoid large residuals.

As a specific example, figure 4.10 shows the density functions of a Gaussian distribution with mean zero and variance $\sigma^2 = 1$, and a Laplace distribution with mean zero and scale parameter $b = 1/\sqrt{2}$ (so that both distributions have the same variance). We have seen in the previous examples that the corresponding penalty functions for these two distributions are the quadratic penalty and the absolute value penalty, respectively. The Gaussian distribution has much smaller tails compared to the Laplace distribution, indicating that large noise values are much less likely to occur under the Gaussian distribution. On the other hand, the Laplace distribution has shaper heads around zero, compared to the Gaussian distribution, indicating that the noise values are more concentrated around zero. As a result, solution of the MLE problem (4.22) under the Gaussian noise tends to avoid large residuals, while solution under the Laplace noise tends to have many residuals equal or very close to zero, which aligns well with our previous observations about the solution characteristics of the two corresponding penalty functions in §4.1.3.

4.2.2 Probabilistic classification

Suppose we are given a dataset (a_i, y_i), $i = 1, \ldots, m$, where $a_i \in \mathbf{R}^n$ are the ith sample feature and $y_i \in \{0, 1\}$ are the corresponding binary class label, generated according to the *logistic model*:

$$\mathbf{prob}(y_i = 1) = p_i, \qquad \mathbf{prob}(y_i = 0) = 1 - p_i,$$

where $p_i \in [0, 1]$ is given by

$$p_i = \frac{\exp(a_i^T x)}{1 + \exp(a_i^T x)} \tag{4.24}$$

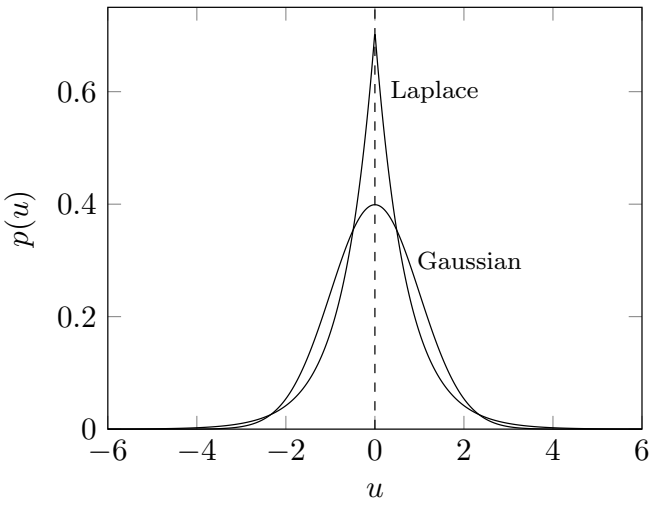

Figure 4.10 Density function of a Gaussian distribution with mean zero and variance $\sigma^2 = 1$, and a Laplace distribution with mean zero and scale parameter $b = 1/\sqrt{2}$.

with some unknown parameter $x \in \mathbf{R}^n$ to be estimated. In practice, the feature vector a_i often has the form

$$a_i = \begin{bmatrix} \tilde{a}_i \\ 1 \end{bmatrix} \in \mathbf{R}^n,$$

where $\tilde{a}_i \in \mathbf{R}^{n-1}$ is the actual feature vector, which is also called the *explanatory variable*, and the last constant entry is added to include an *intercept* or *bias* term in the model.

Logistic models are widely used to model the probability of binary events in various applications, such as medical diagnosis, spam email detection, credit risk assessment, etc. For example, in medical diagnosis, the feature vector may represent a series of the medical test results of a patient, and the label indicates whether the patient has a certain disease or not. In spam email detection, the feature vector may represent various characteristics of an email, such as the presence of certain keywords or the sender's address, and the label indicates whether the email is spam or not. Fitting a logistic model to some dataset aims to classify the samples into two classes based on their features, in the sense of estimating the probability of each class given the features.

The problem of estimating the parameter x with data (a_i, y_i), $i = 1, \ldots, m$, observed from a logistic model is called the *logistic regression*. To formulate it as an MLE problem, we can assume without loss of generality that for this (or the equivalently permuted) dataset, there exists some $k \in \{1, \ldots, m\}$ such that

$$y_i = \begin{cases} 1, & i = 1, \ldots, k \\ 0, & i = k+1, \ldots, m. \end{cases}$$

Then likelihood of x given the label $y \in \mathbf{R}^m$ for all samples is expressed as

$$p_x(y) = \prod_{i=1}^{k} p_i \prod_{i=k+1}^{m} (1 - p_i),$$

where $p_i \in [0, 1]$ is given by the logistic model (4.24). The corresponding log-likelihood function is

$$\begin{aligned} l(x) &= \sum_{i=1}^{k} \log p_i + \sum_{i=k+1}^{m} \log(1 - p_i) \\ &= \sum_{i=1}^{k} \log\left(\frac{\exp(a_i^T x)}{1 + \exp(a_i^T x)}\right) + \sum_{i=k+1}^{m} \log\left(\frac{1}{1 + \exp(a_i^T x)}\right) \\ &= \sum_{i=1}^{k} \left(a_i^T x - \log(1 + \exp(a_i^T x))\right) - \sum_{i=k+1}^{m} \log(1 + \exp(a_i^T x)) \\ &= \sum_{i=1}^{m} \left(y_i a_i^T x - \log(1 + \exp(a_i^T x))\right). \end{aligned}$$

The MLE problem formulation for logistic regression is hence given by

$$\text{maximize} \quad l(x) = \textstyle\sum_{i=1}^{m} \left(y_i a_i^T x - \log(1 + \exp(a_i^T x))\right) \tag{4.25}$$

with variable $x \in \mathbf{R}^n$. The problem (4.25) is a convex optimization problem since l is concave in x (which is not obvious).

Remark 4.4 *Convexity of logistic regression.* To show the convexity of the logistic regression problem (4.25), we need to show that the log-likelihood function

$$l(x) = \sum_{i=1}^{m} \left(y_i a_i^T x - \log(1 + \exp(a_i^T x))\right)$$

is concave in x. We may use the following facts of convex functions from §2.3:

(a) The sum of any number of concave functions is concave.

(b) Affine (and hence, linear) functions are both convex and concave.

(c) The log-sum-exp function

$$f(u) = \log \sum_{i=1}^{k} \exp u_i$$

is convex in $u \in \mathbf{R}^k$.

(d) The composition of a convex function with an affine function is convex.

According to (a), we only need to show that each term in the summation is concave. The first term $y_i a_i^T x$ is linear in x, hence concave by (b). The second term $\log(1 + \exp(a_i^T x))$ can be expressed as

$$\log(1 + \exp(a_i^T x)) = \log\left(\exp(0) + \exp(a_i^T x)\right),$$

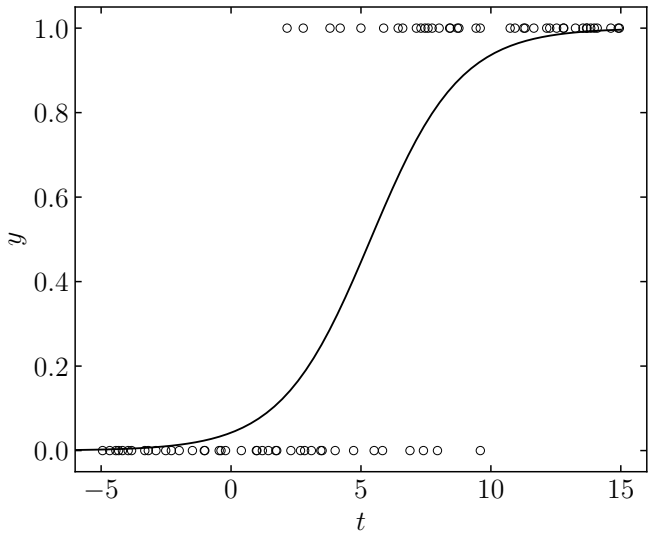

Figure 4.11 *Logistic regression.* Plot of the dataset (a_i, y_i), $i = 1, \ldots, 80$, with $a_i = (t_i, 1)$, shown as circles. The solid curve shows the probability $\mathbf{prob}(y = 1)$ for different values of the explanatory variable t, according to the logistic model with parameter estimated from the dataset.

which can be seen as the composition of the log-sum-exp function and the affine function $x \mapsto (0, a_i^T x)$. Hence, by (c) and (d), this term is convex in x. Since the negative of a convex function is concave, we conclude that each term in the summation is concave, and hence l is concave in x.

Example 4.6 *Logistic regression.* Consider a dataset of $m = 80$ points (a_i, y_i), $i = 1, \ldots, m$, where $a_i = (t_i, 1) \in \mathbf{R}^2$ and $y_i \in \{0, 1\}$ are generated from a logistic model with some unknown parameter $x \in \mathbf{R}^2$. Figure 4.11 plots this dataset on a 2-dimensional plane, where each sample is shown as a circle. Intuitively, we can see that it is more likely for a sample to have label $y = 1$ when its explanatory variable $t > 5$, and vice versa. Besides, when $t < 0$ or $t > 10$, it is very likely that the corresponding samples are labeled as 0 or 1, respectively.

For some explanatory variable value t, the solid curve in figure 4.11 shows the probability

$$\mathbf{prob}(y = 1) = \frac{\exp(x_1^\star t + x_2^\star)}{1 + \exp(x_1^\star t + x_2^\star)}$$

where $x^\star \in \mathbf{R}^2$ is the solution of the logistic regression problem (4.25) with this dataset. This curve represents the estimated probability of a sample being labeled as 1 under the logistic model with parameter $x^\star$, which aligns well with our previous observations about the dataset.

The idea of logistic regression can be generalized to the *multiclass* case, where the label may take values in $\{1, \ldots, K\}$ with $K > 2$; see exercise 4.4.

4.2.3 Counting problems

Consider a dataset (a_i, y_i), $i = 1, \ldots, m$, where $a_i \in \mathbf{R}^n$ are the ith sample feature and $y_i \in \mathbf{Z}_+$ are the corresponding nonnegative integer valued *count labels*, generated according to the *Poisson model*:

$$\mathbf{prob}(y_i = k) = \frac{\lambda_i^k \exp(-\lambda_i)}{k!}$$

for all $k \in \mathbf{Z}_+$, where the mean value $\lambda_i > 0$ is modeled as

$$\log \lambda_i = a_i^T x$$

with some unknown parameter $x \in \mathbf{R}^n$ to be estimated.

Poisson models are widely used to model count data in various applications, such as the number of customer arrivals at a store, the number of emails received in an hour, the number of accidents occurring at a traffic intersection, etc. Fitting a Poisson model to some dataset aims to estimate the relationship between the features and the mean count values.

The problem of estimating the parameter x with data (a_i, y_i), $i = 1, \ldots, m$, observed from a Poisson model is called the *Poisson regression*. To formulate it as an MLE problem, notice that the likelihood of model parameter x given the label $y \in \mathbf{R}^m$ for all samples is expressed as

$$p_x(y) = \prod_{i=1}^m \frac{\lambda_i^{y_i} \exp(-\lambda_i)}{y_i!},$$

where $\lambda_i = \exp(a_i^T x)$ according to the Poisson model. The corresponding log-likelihood function is then expressed as

$$\begin{aligned} l(x) &= \sum_{i=1}^m \left(y_i \log \lambda_i - \lambda_i - \log(y_i!)\right) \\ &= \sum_{i=1}^m \left(y_i a_i^T x - \exp(a_i^T x) - \log(y_i!)\right). \end{aligned}$$

Since the last term $\log(y_i!)$ is constant for all $i = 1, \ldots, m$, it can be removed from the objective without influencing the solution, so the MLE problem formulation for Poisson regression is given by

$$\text{maximize} \quad l(x) = \textstyle\sum_{i=1}^m \left(y_i a_i^T x - \exp(a_i^T x)\right) \tag{4.26}$$

with variable $x \in \mathbf{R}^n$. Similar to logistic regression, the Poisson regression problem (4.26) is also a convex optimization problem.

Figure 4.12 shows an example of Poisson regression. The dataset (a_i, y_i), $i = 1, \ldots, m$, has $m = 500$ samples, where $a_i = (t_i, 1) \in \mathbf{R}^2$ and $y_i \in \mathbf{Z}_+$ are generated from a Poisson model with some unknown parameter $x \in \mathbf{R}^2$.

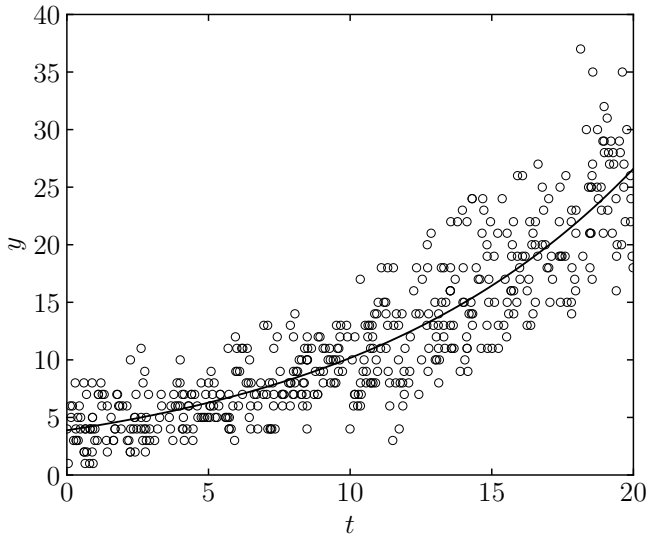

Figure 4.12 *Poisson regression.* Plot of the dataset (a_i, y_i), $i = 1, \ldots, 500$, with $a_i = (t_i, 1)$, shown as circles. The solid curve shows the mean count value $\lambda = \exp(x_1^\star t + x_2^\star)$ for different values of the explanatory variable t, according to the Poisson model with parameter $x^\star \in \mathbf{R}^2$ estimated from the dataset.

4.2.4 Gaussian covariance estimation

Consider a random vector $y \in \mathbf{R}^n$ from a multivariate Gaussian distribution with mean zero and covariance matrix $X = \mathbf{E}\, yy^T$, and $X \in \mathbf{S}^n_{++}$ is positive definite. Suppose we are given a dataset $y_i \in \mathbf{R}^n$, $i = 1, \ldots, m$, which are m independent samples drawn from this distribution, and our goal is to estimate the covariance matrix X from this dataset.

Recall that the probability density function of a multivariate Gaussian distribution with mean zero and covariance matrix X is given by

$$p_X(u) = (2\pi)^{-n/2} (\det X)^{-1/2} \exp\left(-\frac{1}{2} u^T X^{-1} u\right)$$

for any $u \in \mathbf{R}^n$, so the log-likelihood of X given the dataset $y_i \in \mathbf{R}^n$, $i = 1, \ldots, m$, is expressed as

$$\begin{aligned} l(X) &= \sum_{i=1}^m \log p_X(y_i) \\ &= \sum_{i=1}^m \left(-\frac{n}{2}\log(2\pi) - \frac{1}{2}\log\det X - \frac{1}{2} y_i^T X^{-1} y_i\right) \\ &= -\frac{mn}{2}\log(2\pi) - \frac{m}{2}\log\det X - \frac{1}{2}\sum_{i=1}^m y_i^T X^{-1} y_i \end{aligned}$$

$$= -\frac{mn}{2}\log(2\pi) - \frac{m}{2}\log\det X - \frac{1}{2}\,\mathbf{tr}\left(X^{-1}\sum_{i=1}^{m} y_i y_i^T\right),$$

where the last equality follows from the property that

$$\sum_{i=1}^{m} y_i^T X^{-1} y_i = \sum_{i=1}^{m} \mathbf{tr}(y_i^T X^{-1} y_i) = \mathbf{tr}\left(\sum_{i=1}^{m} X^{-1} y_i y_i^T\right) = \mathbf{tr}\left(X^{-1}\sum_{i=1}^{m} y_i y_i^T\right).$$

We now change the variable from X to its inverse $S = X^{-1}$, which is also positive definite since we have assumed that $X \in \mathbf{S}^n_{++}$. Let

$$Y = \frac{1}{m}\sum_{i=1}^{m} y_i y_i^T$$

be the sample covariance matrix of the dataset (and hence $Y \in \mathbf{S}^n_+$ is positive semidefinite), then we have

$$l(S) = -\frac{mn}{2}\log(2\pi) + \frac{m}{2}\log\det S - \frac{m}{2}\,\mathbf{tr}(SY).$$

By removing the constant term and scaling the objective, the MLE problem for estimating the covariance matrix X from the dataset $y_i \in \mathbf{R}^n$, $i = 1, \ldots, m$, is given by

$$\begin{array}{ll} \text{maximize} & \log\det S - \mathbf{tr}(SY) \\ \text{subject to} & S \succ 0 \end{array} \tag{4.27}$$

with variable S. (The constraint $S \succ 0$ is actually implicitly included in the domain of the log-determinant function $\log\det\colon \mathbf{S}^n_{++} \to \mathbf{R}$; we write it explicitly here for clarity.) This problem is a convex optimization problem since the log-determinant function is concave on $\mathbf{S}^n_{++}$ and $\mathbf{tr}(SY)$ is linear in S for fixed Y.

The Gaussian covariance estimation problem (4.27) can be solved analytically. The gradient of the objective function is given by

$$\nabla(\log\det S - \mathbf{tr}(SY)) = S^{-1} - Y.$$

If $Y \in \mathbf{S}^n_{++}$, then setting the gradient to zero gives the unique optimal point

$$S^\star = Y^{-1},$$

so the estimated covariance matrix is

$$X^\star = (S^\star)^{-1} = Y = \frac{1}{m}\sum_{i=1}^{m} y_i y_i^T,$$

which is simply the sample covariance matrix of the dataset (and aligns well with our intuition). If $Y \in \mathbf{S}^n_+$ but is not positive definite, then the problem (4.27) is unbounded above; see exercise 4.5.

4.3 Nonparametric distribution estimation

We consider the problem of estimating a distribution for some discrete random variable $Z \in \{1, \ldots, n\}$ with n possible outcomes. Suppose $x \in \mathbf{R}^n$ with $x_i = \mathbf{prob}(Z = i)$ for all $i = 1, \ldots, n$ is the probability distribution of Z, then it is obvious that x must satisfy the constraints

$$\sum_{i=1}^{n} x_i = 1 \quad \text{and} \quad x_i \geq 0$$

for all $i = 1, \ldots, n$, *i.e.*, x lies in the probability simplex

$$\{x \in \mathbf{R}^n \mid x \succeq 0,\ \mathbf{1}^T x = 1\}.$$

Conversely, any point in the probability simplex in $\mathbf{R}^n$ defines a valid (and unique) probability distribution for the discrete random variable $Z \in \{1, \ldots, n\}$. Therefore, the nonparametric distribution estimation problem for a discrete random variable with n possible outcomes can be expressed as

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \succeq 0, \quad \mathbf{1}^T x = 1, \end{array} \tag{4.28}$$

where the variable $x \in \mathbf{R}^n$ is the probability distribution to be estimated, and $f : \mathbf{R}^n \to \mathbf{R}$ is some objective function that quantifies the goodness of the estimated distribution. Since the inequality and equality constraints are both affine, this is a convex optimization problem if f is convex.

We now introduce several examples of the basic nonparametric distribution estimation problem (4.28) with different choices of the objective function f. Some of these examples may look trivial and not so useful, but we will see later that they appear as important building blocks to represent more complex models.

Maximum entropy estimation

The *maximum entropy distribution* of the random variable $Z \in \{1, \ldots, n\}$ (under no prior information) is defined as the solution of the problem

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{n} x_i \log x_i \\ \text{subject to} & x \succeq 0, \quad \mathbf{1}^T x = 1 \end{array} \tag{4.29}$$

with variable $x \in \mathbf{R}^n$. Since the negative entropy function $\sum_{i=1}^{n} x_i \log x_i$ is convex in x, this is a convex optimization problem. The unique optimal point of this problem is given by $x^\star \in \mathbf{R}^n$ with

$$x_i^\star = \frac{1}{n}$$

for all $i = 1, \ldots, n$, which is the uniform distribution over the n possible outcomes (see exercise 4.6).

The maximum entropy estimation problem (4.29) is more interesting when there are some prior information about x presented as additional constraints, in which case the maximum entropy distribution $x^\star$ corresponds to the most equivocal or most random distribution, among those consistent with the prior information. When there is no prior information, as in (4.29), the uniform distribution is indeed the most random one.

Minimum KL-divergence estimation

Suppose we are given some prior distribution $q \in \mathbf{R}^n$ for the random variable $Z \in \{1, \ldots, n\}$, where $q_i = \mathbf{prob}(Z = i)$ for all $i = 1, \ldots, n$. We may want to estimate a distribution $x \in \mathbf{R}^n$ that is as close as possible to this prior distribution q, which corresponds to minimizing the *KL-divergence* between x and q, *i.e.*,

$$\begin{array}{ll} \text{minimize} & D_{\text{kl}}(x, q) = \sum_{i=1}^n x_i \log(x_i/q_i) \\ \text{subject to} & x \succeq 0, \quad \mathbf{1}^T x = 1 \end{array} \tag{4.30}$$

with variable $x \in \mathbf{R}^n$ (*cf.* (2.16) on page 43). By Gibbs' inequality (see exercise 2.9), it is easily seen that the solution of the problem (4.30) is given by $x^\star = q$. When the prior distribution is uniform, *i.e.*, $q_i = 1/n$ for all $i = 1, \ldots, n$, the KL-divergence $D_{\text{kl}}(x, q)$ reduces to the negative entropy function plus a constant, so the problem (4.30) reduces to the maximum entropy estimation problem (4.29). Indeed, in this case, the minimum KL-divergence solution is the uniform distribution.

Maximum likelihood estimation

Suppose we are given a dataset $z_i \in \{1, \ldots, n\}$, $i = 1, \ldots, m$, which are m independent samples drawn from some unknown distribution of the discrete random variable Z. The log-likelihood of a distribution $x \in \mathbf{R}^n$ given this dataset has the form

$$l(x) = \log \prod_{i=1}^m x_{z_i} = \sum_{i=1}^m \log x_{z_i},$$

where x_{z_i} is the probability of outcome z_i under distribution x. The convexity of this log-likelihood l can be seen from the fact that it is a sum of logarithm functions precomposed with affine functions of x, which is hence concave. The corresponding estimation problem is given by

$$\begin{array}{ll} \text{minimize} & -l(x) = -\sum_{i=1}^m \log x_{z_i} \\ \text{subject to} & x \succeq 0, \quad \mathbf{1}^T x = 1 \end{array} \tag{4.31}$$

with variable $x \in \mathbf{R}^n$.

The problem (4.31) has an analytical solution. Let $c \in \mathbf{R}^n$ be the count vector of the dataset, where for each $i = 1, \ldots, n$, the ith entry c_i is the number of occurrences of outcome i in the dataset, so that $c_1 + \cdots + c_n = m$. Then the objective of (4.31) can be equivalently replaced by

$$-l(x) = -\frac{1}{m} \sum_{i=1}^n c_i \log x_i = -\sum_{i=1}^n p_i \log x_i,$$

where $p = c/m \in \mathbf{R}^n$. Then, by Gibbs' inequality, the unique optimal point of (4.31) is given by

$$x^\star = p = \frac{c}{m},$$

which is simply the empirical distribution of Z according to the dataset.

Bounding problems

We can compute upper or lower bounds on some probability values of interest, or the expected value of some function of the discrete random variable Z, by solving nonparametric distribution estimation problems with appropriate objective functions. For example, to compute an lower bound on the probability $\mathbf{prob}(Z \in S)$ for some subset $S \subseteq \{1, \ldots, n\}$, we can solve the problem

$$\begin{array}{ll} \text{minimize} & \sum_{i \in S} x_i \\ \text{subject to} & x \succeq 0, \quad \mathbf{1}^T x = 1 \end{array}$$

with variable $x \in \mathbf{R}^n$. As another example, to compute an lower bound on the sum of k largest probabilities of Z, we can solve the problem

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{k} x_{[i]} \\ \text{subject to} & x \succeq 0, \quad \mathbf{1}^T x = 1 \end{array}$$

with variable $x \in \mathbf{R}^n$, where $x_{[1]} \geq x_{[2]} \geq \cdots \geq x_{[n]}$ are the entries of x sorted in nonincreasing order. These problems are both convex since their objectives are linear and piecewise linear functions of x, respectively; (see example 2.16). Finally, suppose the random variable Z takes values in $\{c_1, \ldots, c_n\}$ for some given numbers $c_i \in \mathbf{R}$. To compute a lower bound on the expected value of some function $g\colon \mathbf{R} \to \mathbf{R}$ of Z, we can solve the problem

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{n} g(c_i) x_i \\ \text{subject to} & x \succeq 0, \quad \mathbf{1}^T x = 1 \end{array}$$

with variable $x \in \mathbf{R}^n$. Here we take Z to be scalar valued, but the formulation can be readily generalized to vector valued random variables.

Mixture models

One important application of nonparametric distribution estimation is to fit *mixture models* or *hierarchical models*. Here we introduce a basic example about fitting a mixture of linear models to some dataset.

Suppose we have a dataset (a_i, y_i), $i = 1, \ldots, m$, where $a_i \in \mathbf{R}^n$ are the ith sample feature and $y_i \in \mathbf{R}$ are the corresponding observed response values. We want to fit a mixture of K linear regression models to this dataset, where each linear model has its own parameter $x_k \in \mathbf{R}^n$ for $k = 1, \ldots, K$. Let $z_i \in \{z \in \mathbf{R}^K \mid z \succeq 0,\ \mathbf{1}^T z = 1\}$ be the probability distribution over the K linear models for the ith sample (to be estimated), where the kth entry of z_i corresponds to the probability of the ith sample being generated from the kth linear model. These probability

distributions $z_1, \ldots, z_m$ are called the *hidden* or *latent factors* under the context of mixture models, since they are not directly observed from the dataset but need to be estimated together with the model parameters. Let $r_i \in \mathbf{R}^K$ be the residual vector for the ith sample, where the kth entry of r_i is given by

$$r_{ik} = a_i^T x_k - y_i,$$

which is the residual of the ith sample under the kth linear model. Then the inverse problem of this mixture of linear models consists in minimizing the total expected residual over all samples, *i.e.*,

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^m z_i^T r_i \\ \text{subject to} & r_{ik} = a_i^T x_k - y_i, \quad z_i \succeq 0, \quad \mathbf{1}^T z_i = 1 \\ & i = 1, \ldots, m, \quad k = 1, \ldots, K, \end{array}$$

where the variables are $x_1, \ldots, x_K \in \mathbf{R}^n$ and $z_1, \ldots, z_m \in \mathbf{R}^K$ (and $r_1, \ldots, r_m \in \mathbf{R}^K$ is auxiliary). This problem is a biconvex optimization problem, since it is convex in $x_1, \ldots, x_K$ when $z_1, \ldots, z_m$ are fixed, and vice versa.

We will see more details about mixture models and latent factor estimation problems in chapter 8.

4.4 Discrimination

Suppose we are given a dataset consisting of two groups of points $\{x_1, \ldots, x_M\}$ and $\{y_1, \ldots, y_N\}$ in $\mathbf{R}^n$. A *discrimination* problem aims at finding a function $f : \mathbf{R}^n \to \mathbf{R}$ such that

$$f(x_i) > 0 \text{ for all } i = 1, \ldots, M$$

and

$$f(y_i) < 0 \text{ for all } i = 1, \ldots, N,$$

or in other words, the 0-level set $\{x \mid f(x) = 0\}$ of the function f *separates* or *classifies* the two groups of points. Any function f that satisfies these conditions is called a *discrimination function* or *discriminator* for the two groups of points.

If we restrict f to be an affine function of the form

$$f(x) = a^T x - b$$

for some $a \in \mathbf{R}^n$ and $b \in \mathbf{R}$, then the discrimination problem reduces to finding a hyperplane $\{x \mid a^T x - b = 0\}$ that separates the two groups of points, and is hence called a *linear discrimination* problem. This can be formulated as the feasibility problem

$$\begin{array}{ll} \text{find} & (a, b) \\ \text{subject to} & a^T x_i - b > 0, \quad i = 1, \ldots, M \\ & a^T y_i - b < 0, \quad i = 1, \ldots, N, \end{array} \tag{4.32}$$

with variables $a \in \mathbf{R}^n$ and $b \in \mathbf{R}$. Noticing that the strict inequality constraints are homogeneous in (a, b), *i.e.*, if some (a, b) satisfies the constraints, then so does

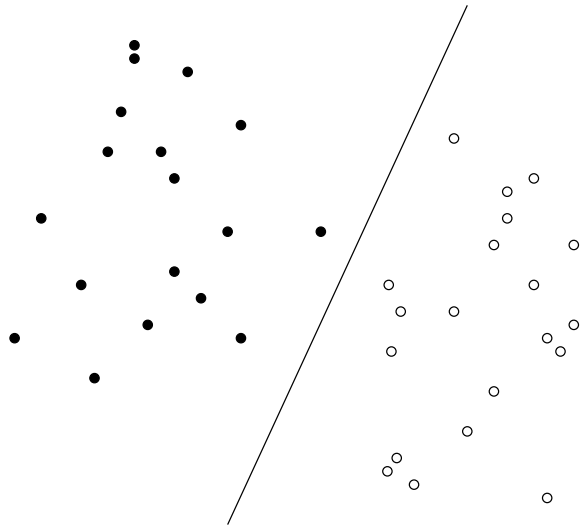

Figure 4.13 *Linear discrimination.* Two groups of points in $\mathbf{R}^2$ shown in open and filled circles, and the 0-level set of a linear discrimination function (shown in solid line) separating them.

$t(a, b)$ for any $t > 0$, the linear discrimination problem (4.32) can be equivalently written as

$$\begin{array}{ll} \text{find} & (a, b) \\ \text{subject to} & a^T x_i - b \geq 1, \quad i = 1, \ldots, M \\ & a^T y_i - b \leq -1, \quad i = 1, \ldots, N \end{array}$$

(see exercise 4.7), which is a linear feasibility problem. Figure 4.13 shows a simple example of two groups of points in $\mathbf{R}^2$ and a linear discrimination function separating them.

If the function f is quadratic, *i.e.*, has the form

$$f(x) = x^T P x + q^T x + r$$

for some $P \in \mathbf{S}^n$, $q \in \mathbf{R}^n$, and $r \in \mathbf{R}$, then the discrimination problem is called a *quadratic discrimination*. The corresponding feasibility problem formulation is given by

$$\begin{array}{ll} \text{find} & (P, q, r) \\ \text{subject to} & x_i^T P x_i + q^T x_i + r > 0, \quad i = 1, \ldots, M \\ & y_i^T P y_i + q^T y_i + r < 0, \quad i = 1, \ldots, N, \end{array} \tag{4.33}$$

which is also a linear feasibility problem (in the variables $P \in \mathbf{S}^n$, $q \in \mathbf{R}^n$, and $r \in \mathbf{R}$). Similar to the linear discrimination problem, the strict inequalities in (4.33) is also homogeneous in (P, q, r), so the problem is equal to

$$\begin{array}{ll} \text{find} & (P, q, r) \\ \text{subject to} & x_i^T P x_i + q^T x_i + r \geq 1, \quad i = 1, \ldots, M \\ & y_i^T P y_i + q^T y_i + r \leq -1, \quad i = 1, \ldots, N. \end{array}$$

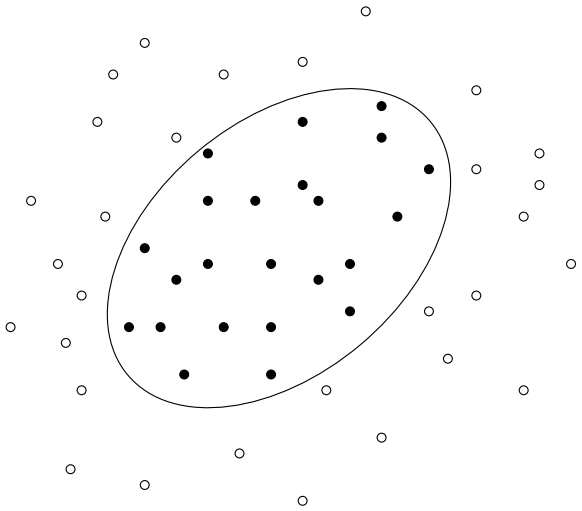

Figure 4.14 *Quadratic discrimination.* Two groups of points in $\mathbf{R}^2$ shown in open and filled circles, and the 0-level set of a quadratic discrimination function (shown in solid ellipsoid) separating them.

For a feasible point (P, q, r) to the problem (4.33), the separating surface $\{x \mid x^T P x + q^T x + r = 0\}$ is a *quadratic surface*. In particular, if $P \succ 0$, then this quadratic surface is an *ellipsoid*; an example is shown in figure 4.14. Therefore, solving the quadratic discrimination problem (4.33) aims at finding a quadratic surface, possibly an ellipsoid, that separates the two groups of points.

Bibliographical notes

The relationship between the log-barrier penalty function (4.16) and the quadratic penalty can be quantified. Specifically, for some residual $r \in \mathbf{R}^m$ with $\|r\|_\infty < \alpha$, the total penalty $\sum_{i=1}^m \phi(r_i)$ is bounded below by $\|r\|_2^2$ and bounded above by $(\phi(\|r\|_\infty)/\|r\|_\infty^2)\|r\|_2^2$; see [BV04, exercise 6.1]. Some applications of this penalty function in robust control (under the name *central* $\mathbf{H}_\infty$*-norm*) can be found in [BB91].

The original idea of quantile penalty function and quantile regression dates back to the 1760s from Boscovich [Sti84]. The book by Koenker [Koe05] provides a systematic review and discussion on this topic.

The Huber penalty function (4.18) and the idea of robust regression was originally introduced by Huber [Hub64, Hub92] in the statistics literature around 1960s, who has also analyzed the robustness properties of approximation problems with different penalty functions; see [HR09]. Lambert-Lacroix and Zwald [LZ12] provides some useful material on the reverse Huber penalty function and its applications.

There exist various smoothed versions of the truncated quadratic penalty function given by (4.19), such as *Welsch/Leclerc penalty* [DW78, Lec89], *Geman-McClure penalty* [GG86], *Cauchy/Lorentzian penalty* [BA96], etc. These and many other similar penalty functions can be represented in a unified form; see [Bar19].

Maximum likelihood estimation is a fundamental statistical estimation method that dates back to the early 20th century from the works of Fisher [Fis22, Fis25]. More recent material on this method can be found in many textbooks on statistics, machine learning, and pattern recognition, such as [DHS00], [Mur12], [BD15], [LM18], [HMC19], [Mur22], [CB24], and [HTZ24].

Some discussion on logistic regression can be found in [HTF09, §4.4]. Poisson regression is discussed in [CWA09] and [CT13]. For more details about the Gaussian covariance estimation problem (4.27), see [And71] and [BV04, page 355].

The concept of *entropy* in information theory and communication was initially introduced by Shannon [Sha48]. The principle of maximum entropy for estimating probability distributions was proposed by Jaynes [Jay57]. More recent discussions on this topic can be found in many textbooks on information theory, *e.g.*, [Mac03] and [Grü07], as well as those on statistics and machine learning listed above.

Mixture models and the problem of mixture decomposition can be traced back to the 1840s [Que46], while Pearson [Pea94], which addresses the problem of decomposing and identifying a mixture of Gaussian distributions, is commonly considered as the first work in fitting mixture models and latent factor estimation. See also the bibliographical notes of chapter 8 for more references on these topics.

In the most general case, the discrimination problems discussed in §4.4 involve determining whether a system of linear inequalities and equalities is feasible. Some theoretical results are discussed under the name of *theorems of the alternative*; see [BV04, §5.8]. A famous example of theorems of the alternative is the *Farkas' lemma* [Far02], which is the best known theorem of alternatives for systems of linear inequalities and equalities. Farkas' lemma can be varied to many further theorems of the alternative by simple modifications; see, *e.g.*, [Gal89, chapter 2], [Man94, §2.4], [Bor13], and [DJ14]. See also the bibliographical notes of chapter 7 for more references on the practical aspects of linear discrimination.

Exercises

Approximation

4.1 *Huber penalty as infimal convolution.* Show that the Huber penalty function $\phi\colon \mathbf{R} \to \mathbf{R}$ defined in (4.18) can be expressed as the *infimal convolution* (see exercise 2.15) of the quadratic penalty and the (scaled) absolute value penalty, *i.e.*,

$$\phi(u) = \inf_{s\in\mathbf{R}} (s^2 + 2M|u-s|)$$

for all $u \in \mathbf{R}$, where $M > 0$ is the parameter of the Huber penalty function.

4.2 *Representative singleton of a set of points.* We consider a special approximation problem given by

$$\text{minimize} \quad \sum_{i=1}^m \|x_i - \mu\|_2^2, \tag{4.34}$$

where the variable is $\mu \in \mathbf{R}^n$, and the problem data are $x_1, \ldots, x_m \in \mathbf{R}^n$. This problem aims at finding a single point μ that best represents the dataset $\{x_1, \ldots, x_m\}$, in the sense that the sum of squared Euclidean distances from the points to the representative point μ is minimized. Express the optimal point $\mu^\star$ of this problem in terms of the dataset $x_1, \ldots, x_m$. What does this optimal point mean geometrically? What is the corresponding optimal value?

4.3 *Weighted representative singleton of a set of points.* Consider the weighted version of the problem (4.34):

$$\text{minimize} \quad \sum_{i=1}^m w_i\|x_i - \mu\|_2^2,$$

where $w \in \mathbf{R}^m_+$ are given weights. Express the optimal point $\mu^\star$ of this problem in terms of the dataset $x_1, \ldots, x_m$ and the weights $w_1, \ldots, w_m$. How should a data point x_i with weight $w_i = 0$ be interpreted in this problem?

Maximum likelihood estimation

4.4 *Multiclass logistic regression.* The *multiclass logistic model* with the number of classes K has the following form: Let $A \in \mathbf{R}^{K\times n}$ be the feature matrix for one sample, where each row of A corresponds to the feature values contributing to one of the K classes. The class label is denoted as $y \in \{e_1, \ldots, e_K\} \subseteq \mathbf{R}^K$, which is *one-hot* (or standard basis vector) encoded, where $e_k \in \mathbf{R}^K$ has 1 in the kth entry and 0 elsewhere. The probability of the class label y taking value e_k for all $k = 1, \ldots, K$ with feature A is given by

$$\mathbf{prob}(y = e_k) = \frac{\exp z_k}{\sum_{i=1}^K \exp z_i}, \qquad z = Ax,$$

where $x \in \mathbf{R}^n$ is the model parameter.

(a) Suppose we are given a dataset (A_i, y_i), $i = 1, \ldots, m$, generated from a multiclass logistic model with some unknown parameter $x \in \mathbf{R}^n$. Formulate the MLE problem for estimating the parameter x with this dataset.

(b) Show that the multiclass logistic regression problem formulated in (a) is convex.
Hint. Use the facts listed in remark 4.4.

4.5 Suppose we are given a dataset $y_1, \ldots, y_m \in \mathbf{R}^n$ observed from a multivariate Gaussian distribution. Consider the Gaussian covariance estimation problem

$$\begin{array}{ll} \text{maximize} & \log\det S - \mathbf{tr}(SY) \\ \text{subject to} & S \succ 0 \end{array}$$

with variable S, and the problem data $Y \in \mathbf{S}^n_+$ given by $Y = 1/m \sum_{i=1}^m y_i y_i^T$ is the sample covariance matrix of the dataset. Show that this problem is unbounded above if Y is not positive definite.
Hint.

- If $Y \in \mathbf{S}^n_+$ is not positive definite, then there exists some nonzero vector $u \in \mathbf{R}^n$ such that $u^T Y u = 0$.
- *Matrix determinant lemma.* For all $u, v \in \mathbf{R}^n$,

$$\det(I + uv^T) = 1 + v^T u.$$

Nonparametric distribution estimation

4.6 *Maximum entropy distribution.* Show that the optimal value of the maximum entropy estimation problem

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^n x_i \log x_i \\ \text{subject to} & x \succeq 0, \quad \mathbf{1}^T x = 1 \end{array}$$

with variable $x \in \mathbf{R}^n$ is $-\log n$ and is achieved at $x^\star = (1/n)\mathbf{1}$.
Hint. Use Gibbs' inequality (see exercise 2.9).

Discrimination

4.7 *Alternative formulation of strict linear discrimination.* Suppose we are given two groups of points $\{x_1, \ldots, x_M\}$ and $\{y_1, \ldots, y_N\}$ in $\mathbf{R}^n$. Show that the linear discrimination conditions

$$a^T x_i - b > 0, \quad i = 1, \ldots, M, \quad \text{and} \quad a^T y_i - b < 0, \quad i = 1, \ldots, N,$$

in the variables $a \in \mathbf{R}^n$ and $b \in \mathbf{R}$ is feasible if and only if the weak inequalities

$$a^T x_i - b \geq 1, \quad i = 1, \ldots, M, \quad \text{and} \quad a^T y_i - b \leq -1, \quad i = 1, \ldots, N,$$

in a and b is feasible.

Chapter 5

Regularization functions

5.1 Multiobjective optimization

5.1.1 Problems with vector-valued objective

So far we have always assumed that the objective function in an optimization problem is scalar-valued. We now extend our discussion to problems with vector-valued objective, *i.e.*, *multiobjective* (or *multicriterion*) optimization problems, defined as

$$\begin{array}{ll} \text{minimize} & f_0(x) = (F_1(x), \ldots, F_k(x)) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \ldots, m \\ & h_i(x) = 0, \quad i = 1, \ldots, p, \end{array} \tag{5.1}$$

where $x \in \mathbf{R}^n$ is the optimization variable, and $f_0 \colon \mathbf{R}^n \to \mathbf{R}^k$ is the vector-valued objective function, with $F_i \colon \mathbf{R}^n \to \mathbf{R}$ being the ith component function of f_0, for $i = 1, \ldots, k$. The (scalar-valued) functions $f_i \colon \mathbf{R}^n \to \mathbf{R}$, $i = 1, \ldots, m$, and $h_i \colon \mathbf{R}^n \to \mathbf{R}$, $i = 1, \ldots, p$, are the inequality and equality constraint functions, respectively. The problem (5.1) is sometimes called a *vector optimization problem*, to distinguish it from the case when f_0 is scalar-valued (which is called a *scalar optimization problem*, in contrast).

We can interpret the components $F_i \colon \mathbf{R}^n \to \mathbf{R}$, $i = 1, \ldots, k$, of the vector-valued objective f_0 as the k different scalar objectives that we want to minimize simultaneously over the variable x, and we refer to F_i as the *ith objective* of the problem (5.1).

We can extend the definition of convex optimization problems to vector optimization as well. The problem (5.1) is called a *convex multiobjective optimization problem*, if for all $x, y \in \mathbf{dom}\, f_0$ and $\theta \in [0, 1]$, we have

$$f_0(\theta x + (1 - \theta)y) \preceq \theta f_0(x) + (1 - \theta) f_0(y), \tag{5.2}$$

and the inequality constraint functions f_i are convex for all $i = 1, \ldots, m$, and the equality constraint functions h_i are affine for all $i = 1, \ldots, p$. In other words, the condition (5.2) requires that for all $i = 1, \ldots, k$, we have

$$F_i(\theta x + (1 - \theta)y) \leq \theta F_i(x) + (1 - \theta) F_i(y),$$

i.e., the ith component F_i of the objective function f_0 is convex.

Interpretation

For scalar optimization problems with objective $f_0 \colon \mathbf{R}^n \to \mathbf{R}$, the meaning of 'minimize $f_0(x)$ over the variable x' is clear: Suppose we have two feasible points x and y, then when we compare their corresponding objective values $f_0(x)$ and $f_0(y)$, there could only be three possible relations, $f_0(x) \leq f_0(y)$, $f_0(x) \geq f_0(y)$, or both, if they are equal. When the first case happens, we can say that the point x is 'better than or equal to' the point y with respect to the objective f_0, so the goal of scalar optimization can be interpreted as finding a point x such that $f_0(x) \leq f_0(y)$ for all feasible points y.

However, for vector optimization problems with objective $f_0 \colon \mathbf{R}^n \to \mathbf{R}^k$, we need to carefully interpret what the objective of (5.1) means. Again consider two feasible points x and y, their corresponding objective values $f_0(x)$ and $f_0(y)$ are now vectors in $\mathbf{R}^k$, which can have the following possible relations when we are trying to compare them:

- $f_0(x) \preceq f_0(y)$, *i.e.*, all components of the vector $f_0(x)$ are smaller than or equal to those of $f_0(y)$;
- $f_0(x) \succeq f_0(y)$, *i.e.*, all components of $f_0(x)$ are larger than or equal to those of $f_0(y)$;
- $f_0(x) = f_0(y)$, where all components of the two vectors are equal;
- neither $f_0(x) \preceq f_0(y)$ nor $f_0(x) \succeq f_0(y)$ holds, which means that some components of $f_0(x)$ are smaller than those of $f_0(y)$, while others are larger, *i.e.*, they are *incomparable*.

In the first three cases, for instance, when $f_0(x) \preceq f_0(y)$, we can still say that the point x is better than or equal to the point y with respect to the (vector-valued) objective f_0, in the sense that x is no worse than y in each ith (scalar) objective F_i for $i = 1, \ldots, k$. In particular, if $F_i(x) \leq F_i(y)$ for all $i = 1, \ldots, k$, and $F_j(x) < F_j(y)$ for at least one j, then we can say that x is unambiguously better than y with respect to the objective f_0.

When the two objective values $f_0(x)$ and $f_0(y)$ are incomparable, on the other hand, we cannot say which point is better, since x could be better than y in some objectives, but worse in others. This, of course, can never happen when the objective f_0 is scalar-valued. As a result, we have to redefine, or extend, the meaning of the keyword 'minimize', when the objective is a vector-valued function, as well as the meaning of optimal values and points for multiobjective optimization problems.

5.1.2 Optimal and Pareto optimal

Optimal values and points

We first consider a special case, in which the meaning of the multiobjective optimization problem (5.1) is clear. If there exists a point $x^\star$ such that

$$f_0(x^\star) \preceq f_0(x) \tag{5.3}$$

for *all* feasible points x, then we call $f_0(x^\star)$ an *optimal value*, and refer to $x^\star$ as an *optimal point*. This requires that, firstly, the objective value $f_0(x^\star) \in \mathbf{R}^k$ must be comparable to *all* other objective values $f_0(x)$ for all feasible points x, and secondly, it must be no larger than any of them in all components, *i.e.*,

$$F_i(x^\star) \leq F_i(x), \quad i = 1, \ldots, k,$$

for all feasible x. In other words, an optimal point $x^\star$ must be simultaneously optimal for each of the scalar optimization problems

$$\begin{array}{ll} \text{minimize} & F_j(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \ldots, m \\ & h_i(x) = 0, \quad i = 1, \ldots, p \end{array}$$

for $j = 1, \ldots, k$.

We can provide optimal values and points of multiobjective optimization problems the following geometric interpretation. Consider the set of objective values achieved by all feasible points of the problem (5.1):

$$\mathcal{O} = \left\{ f_0(x) \in \mathbf{R}^k \;\middle|\; \begin{array}{l} x \in \bigcap_{i=0}^m \mathbf{dom}\, f_i \cap \bigcap_{i=1}^p \mathbf{dom}\, h_i \\ f_i(x) \leq 0, \quad i = 1, \ldots, m \\ h_i(x) = 0, \quad i = 1, \ldots, p \end{array} \right\},$$

which is called the set of *achievable objective values*. A point $x^\star$ is optimal if and only if it is feasible and

$$\mathcal{O} \subseteq f_0(x^\star) + \mathbf{R}^k_+.$$

The set on the right-hand side can be interpreted as the set of all objective values that are no smaller than (*i.e.*, worse than, or equal to) $f_0(x^\star)$ in all components (but is not necessarily achievable), so according to the condition (5.3), to make $x^\star$ optimal, the set $\mathcal{O}$ of achievable objective values must be contained in this set.

This geometric interpretation is illustrated in figure 5.1. The set $\mathcal{O}$ of achievable objective values for some multiobjective optimization problem in $\mathbf{R}^2$ is shown darker. In the figure, the point labeled $f_0(x^\star)$ is the optimal value of the problem, and $x^\star$ is an optimal point, since the objective value $f_0(x^\star)$ can be compared to every other achievable value $f_0(x)$, and is better than or equal to (which means, in the figure, 'is below and to the left of') all possible $f_0(x)$.

Most multiobjective optimization problems do not have an optimal value and an optimal point, but this does occur in some special cases. When a multiobjective optimization problem have an optimal value, then it is unique. In this case, an optimal point is, roughly speaking, unambiguously a best choice among all feasible x for the problem (5.1).

Pareto optimal values and points

We now consider the case where the problem (5.1) does not have an optimal point or optimal value, which occurs in most multiobjective optimization problems of

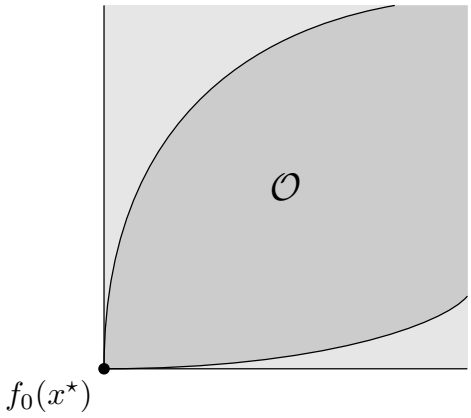

Figure 5.1 The darker region shows the set $\mathcal{O}$ of achievable objective values for a multiobjective optimization problem in $\mathbf{R}^2$. The point $f_0(x^\star)$ (shown circle) is the optimal value of the problem, and the lightly shaded region shows the set $f_0(x^\star) + \mathbf{R}^2_+$.

interest. We say that a feasible point $x^{\rm po} \in \mathbf{R}^n$ is *Pareto optimal*, or *efficient*, if for any feasible $x \in \mathbf{R}^n$, the relation

$$f_0(x) \preceq f_0(x^{\rm po}) \tag{5.4}$$

implies that $f_0(x) = f_0(x^{\rm po})$, *i.e.*, any feasible point x that is better than or equal to $x^{\rm po}$ in all objectives has exactly the same objective value as $x^{\rm po}$. The condition (5.4) can be reexpressed as the following: If x is feasible and satisfies

$$F_i(x) \leq F_i(x^{\rm po}), \quad i = 1, \ldots, k,$$

then $F_i(x) = F_i(x^{\rm po})$ for all $i = 1, \ldots, k$. In other words, a feasible point is Pareto optimal if and only if there is no other feasible point that can improve all the objectives simultaneously. We refer to the objective value $f_0(x^{\rm po})$ of a Pareto optimal point $x^{\rm po}$ as a *Pareto optimal value* for the multiobjective problem (5.1).

We can also provide Pareto optimal values and points a geometric interpretation. A point $x^{\rm po}$ is Pareto optimal if and only if it is feasible and

$$\mathcal{O} \cap (f_0(x^{\rm po}) - \mathbf{R}^k_+) = \{f_0(x^{\rm po})\}.$$

The set $f_0(x^{\rm po}) - \mathbf{R}^k_+$ can be interpreted as the set of all objective values that are no larger than (*i.e.*, better than or equal to) $f_0(x^{\rm po})$ in all components, so the condition (5.4) requires that the set $\mathcal{O}$ of achievable objective values can only intersect this set at the single point $f_0(x^{\rm po})$ itself. It is readily seen that a multiobjective optimization problem may have many Pareto optimal values and points, which form the so-called *Pareto front* of the problem.

These interpretations are illustrated in figure 5.2. The darker region draws the set $\mathcal{O}$ of achievable objective values for some multiobjective optimization problem in $\mathbf{R}^2$. It is easily seen that this problem does not have an optimal value, since

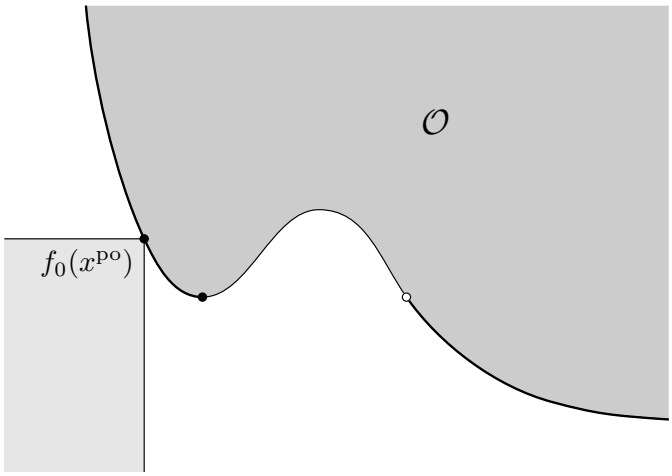

Figure 5.2 Example of the set $\mathcal{O}$ of achievable objective values in $\mathbf{R}^2$ (shown darker). The point $f_0(x^{\mathrm{po}})$ (shown dot) is a Pareto optimal value of the problem, and the lightly shaded region shows the set $f_0(x^{\mathrm{po}}) - \mathbf{R}^2_+$. The thick curve on the boundary of $\mathcal{O}$ is the Pareto front. Note that the end point on the left (shown dot) is included in the Pareto front, while the one on the right (shown circle) is not.

there is no point in $\mathcal{O}$ that can be compared to all other points in $\mathcal{O}$ and is better than or equal to all of them. However, this problem has many Pareto optimal values and points, which form the Pareto front shown as the thick curve on the lower left boundary of $\mathcal{O}$. As one example, the point labeled $f_0(x^{\mathrm{po}})$ is a Pareto optimal value of the problem, and x^{po} is a Pareto optimal point, since the only achievable objective value that is better than or equal to $f_0(x^{\mathrm{po}})$ is $f_0(x^{\mathrm{po}})$ itself.

5.1.3 Scalarization

Scalarization is a standard technique for finding Pareto optimal points of multiobjective optimization problems. Let $\lambda \in \mathbf{R}^k$ and $\lambda \succ 0$, and consider the following scalar optimization problem:

$$\begin{array}{ll} \text{minimize} & \lambda^T f_0(x) = \sum_{i=1}^k \lambda_i F_i(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \ldots, m \\ & h_i(x) = 0, \quad i = 1, \ldots, p \end{array} \tag{5.5}$$

with variable $x \in \mathbf{R}^n$, where the functions f_i, $i = 0, \ldots, m$, and h_i, $i = 1, \ldots, p$, are the same as those in the original multiobjective optimization problem (5.1). Note that the problem (5.5) is not necessarily convex, but this does not affect the discussions below.

For each fixed $\lambda \succ 0$, if x is an optimal point of the problem (5.5), then x is a Pareto optimal point of the original multiobjective optimization problem (5.1). (Although in practice, solving the general scalarized problem (5.5) could be very

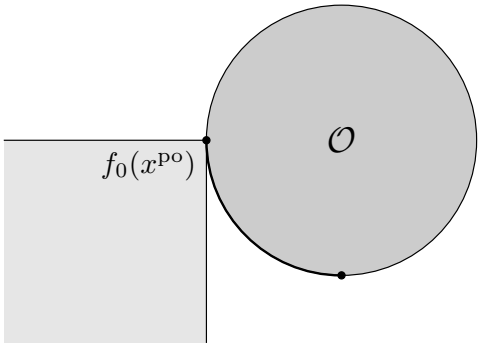

Figure 5.3 An example of the set $\mathcal{O}$ of achievable objective values in $\mathbf{R}^2$ (shown darker). The thick curve on the lower left boundary of $\mathcal{O}$ is the Pareto front, and the two points at the ends of the Pareto front (shown circle) are Pareto optimal values that cannot be obtained from scalarization with $\lambda \succ 0$.

difficult.) To show this, suppose x is optimal for (5.5), but is not Pareto optimal for (5.1). Then there exists some feasible point y of (5.1) such that

$$f_0(y) \preceq f_0(x) \quad \text{and} \quad f_0(y) \neq f_0(x).$$

This implies that $f_0(x) - f_0(y) \succeq 0$ and is nonzero, so we have

$$\lambda^T(f_0(x) - f_0(y)) > 0,$$

i.e., $\lambda^T f_0(y) < \lambda^T f_0(x)$, which contradicts the assumption that x is optimal for (5.5).

The vector λ is a (hyper)parameter of the scalar optimization problem (5.5), and is some times called the *weight vector*. By varying the weight vector $\lambda \succ 0$ and solving the scalarized problem, we can obtain (possibly) different Pareto optimal points of the original multiobjective problem (5.1). Note that, however, not all Pareto optimal points of (5.1) can be obtained by solving (5.5) for some $\lambda \succ 0$. Figure 5.3 shows an example of this situation, where the Pareto optimal point at the two ends of the Pareto front (shown thicker) cannot be obtained from scalarization with $\lambda \succ 0$.

This method of scalarization can be interpreted geometrically as follows. Notice that for each fixed $\lambda \succ 0$, a point $x \in \mathbf{R}^n$ is optimal for the problem (5.5) if and only if it is feasible and $\lambda^T f_0(x) \leq \lambda^T f_0(y)$ for all feasible $y \in \mathbf{R}^n$, *i.e.*,

$$\lambda^T(z - f_0(x)) \geq 0$$

for all $z \in \mathcal{O}$. This implies that the hyperplane $\{z \in \mathbf{R}^k \mid \lambda^T z = \lambda^T f_0(x)\}$ supports the set $\mathcal{O}$ at the point $f_0(x)$, with normal vector $\lambda \in \mathbf{R}^k_{++}$ pointing inward to the set $\mathcal{O}$. From this interpretation we can also see that if $f_0(x)$ is a Pareto optimal value of the problem (5.1), but there exists no supporting hyperplane of $\mathcal{O}$ at the point $f_0(x)$ with positive inward normal vector, then there is no $\lambda \succ 0$ such that the

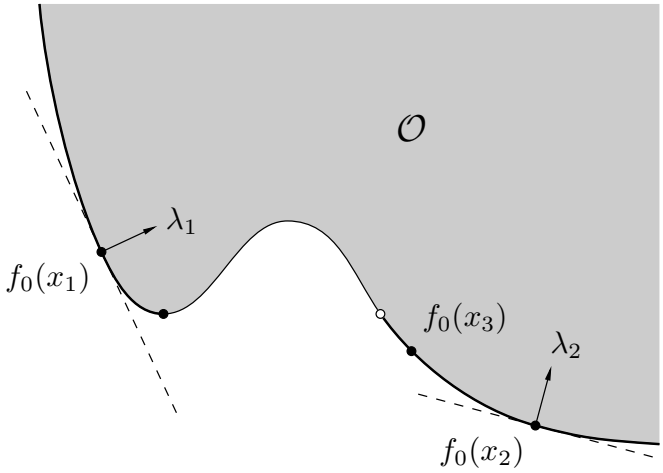

Figure 5.4 The shaded region shows an example of the set $\mathcal{O}$ of achievable objective values in $\mathbf{R}^2$. The thick curve on the lower boundary of $\mathcal{O}$ is the Pareto front. The dashed lines are the supporting hyperplanes of $\mathcal{O}$ obtained by solving the scalarized problem (5.5) with weight vectors λ_1 and λ_2, respectively, which intersect with $\mathcal{O}$ at the Pareto optimal values $f_0(x_1)$ and $f_0(x_2)$. The point $f_0(x_3)$ is a Pareto optimal value but cannot be obtained from scalarization.

corresponding Pareto optimal point x is optimal to the scalarized problem (5.5), *i.e.*, this Pareto optimal point cannot be obtained from scalarization.

The discussions presented above is illustrated in figure 5.4. The shaded region shows the set $\mathcal{O}$ of achievable objective values for some multiobjective optimization problem in $\mathbf{R}^2$. By taking the weight vector as λ_1 and λ_2 and solving the scalarized problem (5.5), we obtain the two supporing hyperplanes (shown dashed) of the set $\mathcal{O}$. The points $f_0(x_1)$ and $f_0(x_2)$ where these hyperplanes touch the set $\mathcal{O}$ are Pareto optimal values of the original multiobjective optimization problem, and x_1, x_2 are the corresponding Pareto optimal points. The Pareto optimal value $f_0(x_3)$ (and the corresponding Pareto optimal point x_3) cannot be obtained by solving the scalarized problem (5.5) for any $\lambda \succ 0$, since there is no supporting hyperplane of $\mathcal{O}$ at the point $f_0(x_3)$.

Scalarization of convex problems

If the multiobjective optimization problem (5.1) is convex, then the scalarized problem (5.5) is also a convex optimization problem for each fixed $\lambda \succ 0$, since the objective function $\lambda^T f_0(x) = \sum_{i=1}^k \lambda_i F_i(x)$ is a nonnegative weighted sum of the convex functions $F_1, \ldots, F_k$. Then it follows immediately that we can now easily find Pareto optimal points of the convex multiobjective problem (5.1) by solving the convex program (5.5) with different weight vectors $\lambda \succ 0$.

In this case, actually, we have the following stronger result that serves as a partial converse to the earlier statement: If $x \in \mathbf{R}^n$ is a Pareto optimal point of

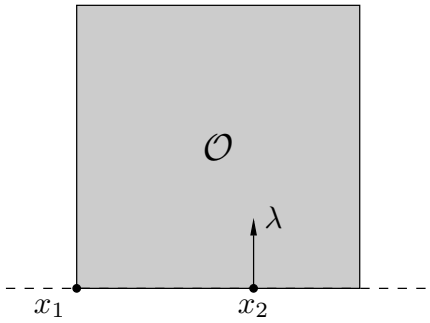

Figure 5.5 An example of the set $\mathcal{O}$ of achievable objective values in $\mathbf{R}^2$ (shown shaded). Points on the bottom boundary of the set $\mathcal{O}$ through x_2 are all solutions of the scalarized problem with $\lambda = (0, 1)$, but only x_1 is Pareto optimal.

the convex multiobjective problem (5.1), then there exists some nonzero $\lambda \succeq 0$ such that x is an optimal point of the corresponding scalarized problem (5.5). In other words, for convex multiobjective optimization problems, all Pareto optimal points can be obtained by solving the scalarized problem (5.5) with some weight vector $\lambda \succeq 0$ that is nonzero.

Remark 5.1 We need to be careful about the conditions on the weight vector λ here. The converse of the statement above is *not* true, *i.e.*, not every solution of the scalarized problem (5.5) with some $\lambda \succeq 0$ ($\lambda \neq 0$) is necessarily a Pareto optimal point of the original multiobjective problem (5.1). Figure 5.5 shows a counterexample in $\mathbf{R}^2$, where all points on the bottom boundary of the set $\mathcal{O}$ (shown shaded) through the point x_2 are solution of the scalarized problem with weight vector $\lambda = (0, 1)$, but only the leftmost point x_1 is Pareto optimal (which is, in this case, also the optimal point).

On the other hand, it is true that *every* solution of the scalarized problem with some $\lambda \succ 0$ is Pareto optimal, but not all Pareto optimal points of the convex multiobjective problem (5.1) can be obtained by solving the scalarized problem with some $\lambda \succ 0$ (even if the original problem is convex). Examples of this situation are already shown in figure 5.3.

These properties can sometimes be helpful (both conceptually and practically) in finding the set of *all* Pareto optimal points of convex multiobjective optimization problems. First of all, by solving the scalarized problem (5.5) with different weight vectors $\lambda \succ 0$, we can obtain a set of Pareto optimal points. Then, we can check whether there are any other Pareto optimal points that could be obtained from scalarization with $\lambda \succeq 0$ ($\lambda \neq 0$). Specifically, for each nonzero $\lambda \succeq 0$, we first solve the scalarized problem (5.5) to obtain an optimal point x, and then check whether x is actually Pareto optimal for the original multiobjective problem (5.1). Note that, in many cases, such Pareto optimal points turn out to be the limits of Pareto optimal points obtained from scalarization with $\lambda \succ 0$. Some simple examples of these 'extreme' Pareto optimal points can be found in figure 5.3, where they appear as the two endpoints of the Pareto front.

5.1.4 Trade-off analysis

Now we give more interpretations and discussions about comparing, or the *trade-off*, between two feasible points of a multiobjective problem whose objective values are incomparable (in particular, two Pareto optimal points, since otherwise it is foolish to accept such points). Let x and y be Pareto optimal points of the multiobjective optimization problem (5.1), where neither $f_0(x) \preceq f_0(y)$ nor $f_0(x) \succeq f_0(y)$ holds. In particular, suppose there are three index sets $A, B, C \subseteq \{1, \ldots, k\}$ such that

$$A \cup B \cup C = \{1, \ldots, k\} \quad \text{and} \quad A \cap B = A \cap C = B \cap C = \emptyset,$$

and we have

$$\begin{aligned} F_i(x) &< F_i(y), \quad i \in A \\ F_i(x) &= F_i(y), \quad i \in B \\ F_i(x) &> F_i(y), \quad i \in C, \end{aligned}$$

so the sets A, B, and C can be interpreted as the sets of objectives F_i for which x performs better than y, for which x and y perform equally, and for which x performs worse than y, respectively. There are two possible situations regarding the relations between the sets A and C:

- If $A = C = \emptyset$, then $F_i(x) = F_i(y)$ for all $i = 1, \ldots, k$, so the two points x and y perform equally in all objectives.
- If either A or C is nonempty, then both A and C must be nonempty (since otherwise one of the two points would be better than or equal to the other), so there are some objectives for which x performs better than y, and some for which x performs worse than y.

In other words, when comparing two Pareto optimal points, they either performs exactly the same in all objectives, or each of them performs better than the other in at least one objective. In the second case, by comparing the point x to y, we say that we have *traded-off* better performance in some objectives F_i with $i \in A$ for worse performance in some other objectives F_j with $j \in C$.

The process of *trade-off analysis* of a multiobjective optimization problem extends the comparison above to general cases, which refers to the study of how much worse we must accept in one or more objectives in order to gain some improvement in other objectives. In other words, it studies what sets of objective values for a multiobjective optimization problem are achievable. For this reason, the Pareto front (*i.e.*, the set of Pareto optimal points) of a multiobjective optimization problem is also called the (optimal) *trade-off surface* (or *trade-off curve* if $k = 2$) of the problem.

Biobjective optimization

As a basic example of trade-off analysis, consider a multiobjective optimization problem in the form (5.1) with only two objectives F_1 and F_2, *i.e.*, $k = 2$, which is also called a *biobjective* problem. Note that in this case, if u and v are two Pareto optimal points with different objective values, then the set B in the discussion

above must be empty, so we have either $F_1(u) < F_1(v)$ and $F_2(u) > F_2(v)$, or $F_1(u) > F_1(v)$ and $F_2(u) < F_2(v)$.

Suppose x is a Pareto optimal point of this biobjective problem, and has objective value $f_0(x) = (F_1(x), F_2(x))$. In trade-off analysis, we might be interested in the following two questions:

- If we want to improve the objective F_1 by some amount $\delta > 0$ from the point x, *i.e.*, to achieve an objective value $F_1(x) - \delta$, then how much worse do we have to accept in the objective F_2?
- Conversely, if we can accept to worsen the objective F_1 by some amount $\delta > 0$ from x, *i.e.*, allows at most $F_1(x) + \delta$, then how much improvement can we gain in the objective F_2?

In the first situation, if a large increase in F_2 must be accepted to realize a small decrease in F_1, we say that there is a *strong trade-off* between the objectives, near the Pareto optimal value $(F_1(x), F_2(x))$; if, on the other hand, a large decrease in F_1 can be obtained with only a small increase in F_2, we say that the trade-off between the objectives is *weak* (near $(F_1(x), F_2(x))$). Similar ideas apply to the second situation as well.

To answer these questions, we can consider the following optimization problem:

$$\begin{array}{ll} \text{minimize} & F_2(y) \\ \text{subject to} & F_1(y) \leq F_1(x) + \delta \\ & f_i(y) \leq 0, \quad i = 1, \ldots, m \\ & h_i(y) = 0, \quad i = 1, \ldots, p, \end{array} \tag{5.6}$$

where $y \in \mathbf{R}^n$ is the variable, $F_1(x)$ is the first objective value of the given Pareto optimal point x, and $\delta \in \mathbf{R}$ is a given parameter. When choosing $\delta < 0$, the optimal value of the problem (5.6), say, $F_2(y^{\mathrm{po}-})$, gives the best value of F_2 that can be achieved when requiring F_1 to improve by at least $|\delta|$ from $F_1(x)$; when choosing $\delta > 0$, the optimal value $F_2(y^{\mathrm{po}+})$ of this problem gives the best value of the objective F_2 that can be achieved when allowing F_1 to worsen by at most δ from $F_1(x)$. It is expected that $F_2(y^{\mathrm{po}-}) \geq F_2(x)$, and $F_2(y^{\mathrm{po}+}) \leq F_2(x)$, so the gaps $F_2(y^{\mathrm{po}-}) - F_2(x)$ and $F_2(x) - F_2(y^{\mathrm{po}+})$ quantitatively measures the trade-off between the two objectives near the Pareto optimal value $f_0(x)$.

The problem (5.6) has a nice geometric interpretation. Let $\mathcal{O}$ be the set of achievable objective values of the vector optimization problem (5.1) with $k = 2$, *i.e.*, $f_0(x) = (F_1(x), F_2(x))$, then the problem (5.6) is equivalent to

$$\begin{array}{ll} \text{minimize} & F_2(y) \\ \text{subject to} & F_1(y) \leq F_1(x) + \delta \\ & (F_1(y), F_2(y)) \in \mathcal{O}, \end{array}$$

where the variable is $(F_1(y), F_2(y)) \in \mathbf{R}^2$. Geometrically, the first inequality constraint defines a halfspace in $\mathbf{R}^2$, where the corresponding boundary hyperplane

$$\{(F_1(y), F_2(y)) \mid F_1(y) = F_1(x) + \delta\}$$

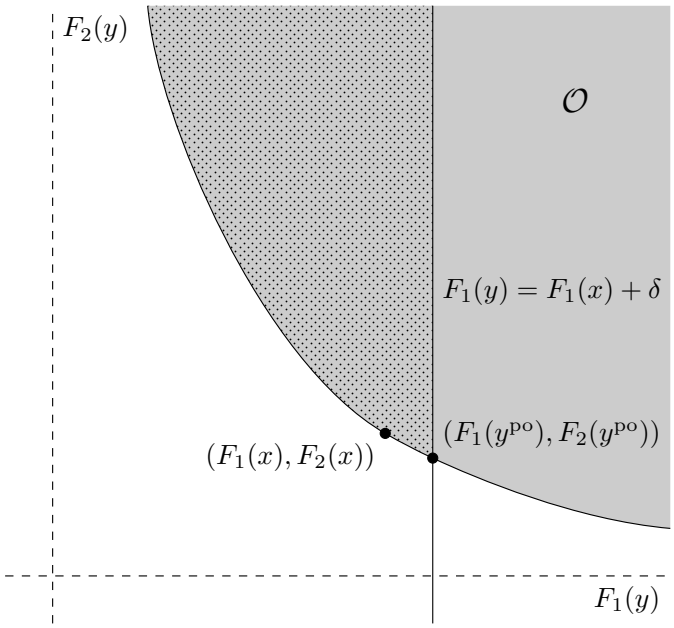

Figure 5.6 Geometric interpretation of the problem (5.6) for trade-off analysis in biobjective optimization. The shaded region shows the set $\mathcal{O}$ of achievable objective values in $(F_1(y), F_2(y))$. The hyperplane $\{(F_1(y), F_2(y)) \mid F_1(y) = F_1(x) + \delta\}$ is shown solid, and the intersection of the halfspace defined by the inequality $F_1(y) \leq F_1(x) + \delta$ and the set $\mathcal{O}$ is shown dotted. The point $(F_1(x), F_2(x))$ is the Pareto optimal value of the given reference point x. The point $(F_1(y^{\mathrm{po}}), F_2(y^{\mathrm{po}}))$ is the optimal value of the problem (5.6) with some $\delta > 0$, and is also a Pareto optimal value of the original biobjective optimization problem.

is orthogonal to the $F_1(y)$-axis, so this problem consists in finding a point in the intersection of this halfspace and the set $\mathcal{O}$, that has the smallest second component. This interpretation is illustrated in figure 5.6.

Scalarization interpretation

Recall that when we scalarize the multiobjective optimization problem (5.1) with weight vector $\lambda \succ 0$, we form the objective

$$\lambda^T f_0(x) = \sum_{i=1}^{k} \lambda_i F_i(x).$$

In this case, the weight λ_i associated with each objective F_i can be interpreted as the *importance* of this objective in the overall performance measure $\lambda^T f_0(x)$. If we have a high desire to make the objective F_i small, then we should choose a large weight λ_i for this objective; if, on the other hand, we are not very concerned about the objective F_j being large, then we can choose a small weight λ_j for it. In other words, by adjusting the weights $\lambda \succ 0$, we are exploring different trade-offs between the objectives of a multiobjective optimization problem.

The ratio λ_i/λ_j of the scalarization weights associated with two objectives F_i and F_j can be interpreted as the *relative importance* of the ith objective compared to the jth objective. Alternatively, this ratio can also be considered as the *exchange rate* between the two objectives F_i and F_j, since it reflects how much worse we are willing to accept (maximally) in the objective F_j in order to gain some improvement in the objective F_i. A simple example illustrates this idea.

Example 5.1 *Exchange rate in biobjective optimization.* Consider a biobjective optimization problem with objectives F_1 and F_2, and suppose we choose the weight vector $\lambda = (\lambda_1, \lambda_2) \succ 0$ for scalarization. Suppose x and y ($x \neq y$) are two Pareto optimal points that has the same scalarized objective value, *i.e.*,

$$\lambda_1 F_1(x) + \lambda_2 F_2(x) = \lambda_1 F_1(y) + \lambda_2 F_2(y). \tag{5.7}$$

We may assume without loss of generality that the corresponding Pareto optimal values of these two points satisfy $F_1(y) < F_1(x)$ and $F_2(y) > F_2(x)$. An example of this situation is shown in figure 5.7.

Let $\delta > 0$ be the improvement in the objective F_1 from x to y, *i.e.*, $F_1(y) = F_1(x) - \delta$, then we have

$$\begin{aligned} \lambda_1 F_1(y) + \lambda_2 F_2(y) &= \lambda_1 (F_1(x) - \delta) + \lambda_2 F_2(y) \\ &= \lambda_1 F_1(x) + \lambda_2 (F_2(y) - (\lambda_1/\lambda_2)\delta) \\ &= \lambda_1 F_1(x) + \lambda_2 F_2(x), \end{aligned}$$

which implies that $F_2(y) = F_2(x) + (\lambda_1/\lambda_2)\delta$. In other words, under this scalarization weight λ, in order to achieve an improvement of δ in the objective F_1 from x to y, we have to accept a worsening of $(\lambda_1/\lambda_2)\delta$ in the objective F_2.

The assumption in (5.7) holds if the Pareto front of the biobjective optimization problem is piecewise affine. In the most general case when this condition does not hold, but the Pareto front is smooth (*e.g.*, shown in figure 5.6), we may still expect that

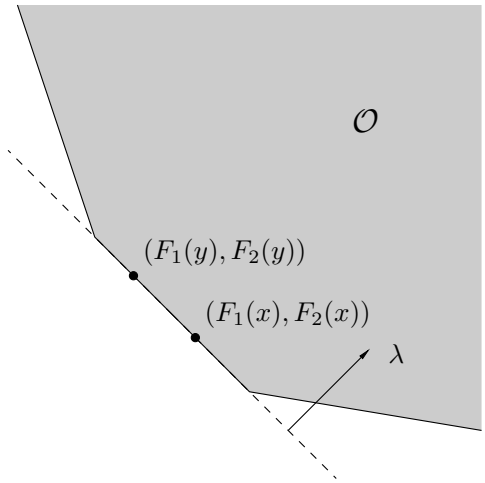

Figure 5.7 Example of two Pareto optimal points $x, y \in \mathbf{R}^2$ that have the same scalarized objective value under some weight vector $\lambda \succ 0$. The shaded region shows the set $\mathcal{O}$ of achievable objective values, and the dashed line is the supporting hyperplane of $\mathcal{O}$ defined by the inward normal vector λ.

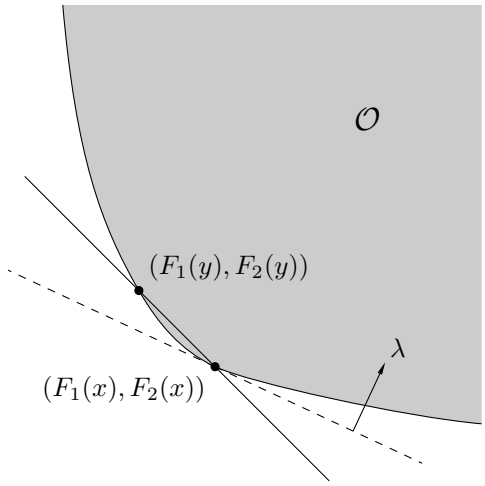

Figure 5.8 The set $\mathcal{O}$ of achievable objective values in $\mathbf{R}^2$ (shown shaded) with a smooth Pareto front. The objective values of two Pareto optimal points x and y are shown as dots. The hyperplane with inward normal vector λ that supports the set $\mathcal{O}$ at the point $(F_1(x), F_2(x))$ is shown dashed. As $(F_1(y), F_2(y))$ approaches $(F_1(x), F_2(x))$, the line segment connecting the two Pareto optimal points converges to this supporting hyperplane.

the discussions above is approximately true. To see this, notice that the scalarization weight λ is a inward normal of the supporting hyperplane (which is a line in $\mathbf{R}^2$) of the set of achievable objective values at (say) $(F_1(x), F_2(x))$, so the slope of this supporting hyperplane is given by $-\lambda_1/\lambda_2$. When $\delta = F_1(x) - F_1(y) \to 0$, the line segment connecting the two Pareto optimal points $(F_1(x), F_2(x))$ and $(F_1(y), F_2(y))$ converges to this supporting hyperplane, which implies that

$$\frac{F_2(y) - F_2(x)}{F_1(y) - F_1(x)} \approx -\frac{\lambda_1}{\lambda_2},$$

i.e., $F_2(y) - F_2(x) \approx (\lambda_1/\lambda_2)\delta$ for small δ. This intuition is shown in figure 5.8. In other words, when the Pareto front is smooth, the scalarization vector $\lambda \succ 0$ provides the local trade-offs among objectives.

As a specific example of how scalarization can be used for trade-off analysis, suppose the weight vector $\lambda \succ 0$ results in the Pareto optimal point x of some multiobjective optimization problem, with objective value $f_0(x) = (F_1(x), \ldots, F_k(x))$. To find a (possibly) different Pareto optimal point that has smaller value in the jth objective, in the price of worsening (some) other objectives, we may choose a new weight vector $\tilde{\lambda} \succ 0$ such that

$$\tilde{\lambda}_j > \lambda_j \quad \text{and} \quad \tilde{\lambda}_i = \lambda_i$$

for all $i \neq j$, $i = 1, \ldots, k$, *i.e.*, we increase the weight on the jth objective. Then solving the scalarized problem with the new weight vector $\tilde{\lambda}$ may give us a different Pareto optimal point $\tilde{x}$, with objective value $F_j(\tilde{x}) \leq F_j(x)$ (and usually, $F_j(\tilde{x}) < F_j(x)$), which improves the jth objective compared to x.

5.2 Regularized approximation

5.2.1 Problem formulation

Consider a system of equations $f(x) = b$ in the variable $x \in \mathbf{R}^n$, where $f\colon \mathbf{R}^n \to \mathbf{R}^m$ and $b \in \mathbf{R}^m$ are given. *Regularized approximation* problems have the general form

$$\text{minimize} \quad (\phi_1(f(x) - b),\ \phi_2(x)) \tag{5.8}$$

with variable $x \in \mathbf{R}^n$, where $f(x) - b$ is the *residual* of the system of equations, the functions $\phi_1\colon \mathbf{R}^m \to \mathbf{R}$ and $\phi_2\colon \mathbf{R}^n \to \mathbf{R}$ measure the size of the approximation residual and the variable x itself, respectively. In the most general case, these two functions ϕ_1 and ϕ_2 can be some norm $\|\cdot\|$ (on $\mathbf{R}^m$ and $\mathbf{R}^n$, respectively), or in the form of some penalty functions (see §4.1.3 and §4.1.4).

The problem (5.8) is a biobjective optimization problem, which is convex if both ϕ_1, ϕ_2 are convex and f is affine. By solving the problem (5.8), we aim at finding a vector x that is small in the sense of $\phi_2(x)$, while also making the residual $f(x) - b$ small in the sense of $\phi_1(f(x) - b)$. (The idea of regularized approximation can be readily extended to settings with three or more objectives; see exercise 5.2.)

Trade-off analysis

We can provide a basic trade-off analysis to the regularized approximation problem (5.8), based on the scalarization

$$\text{minimize} \quad \lambda_1\phi_1(f(x)-b)+\lambda_2\phi_2(x) \tag{5.9}$$

with variable $x \in \mathbf{R}^n$, where $\lambda \succ 0$ is a hyperparameter of this problem that controls the trade-off between the two objectives $\phi_1(f(x)-b)$ and $\phi_2(x)$. Specifically, the ratio λ_2/λ_1 represents the relative importance of finding a small vector x compared to finding a good approximation of the system of equations $f(x)=b$. By varying the hyperparameter $\lambda \in \mathbf{R}^2_{++}$, we can obtain the optimal trade-off curve (*i.e.*, the Pareto front) of the problem (5.8), except two extreme points that correspond to the limits $\lambda_2 \to 0$ and $\lambda_1 \to 0$, respectively.

Usually, in practice, the first extreme Pareto optimal point can be obtained by solving the problem

$$\text{minimize} \quad \phi_1(f(x)-b)$$

with variable $x \in \mathbf{R}^n$ (which is simply the problem (5.9) with $\lambda=(1,0)$). The solution of this problem is the best possible approximation of the system of equations $f(x)=b$ under the residual measure ϕ_1.

Another extreme Pareto optimal point (usually) corresponds to the solution of the problem

$$\text{minimize} \quad \phi_2(x)$$

which is just the problem of finding the smallest possible vector x in the sense of $\phi_2(x)$, without considering the approximation residual at all. In most cases, the optimal of this problem is achieved at $x=0$, and the corresponding value of the first objective in (5.8) is $\phi_1(b)$ (assuming ϕ_1 is symmetric), which is simply the size of the vector b measured by the function ϕ_1, and is the worst possible value of the first objective ϕ_1 that can be achieved by any Pareto optimal point x of the problem (5.8).

Relative scalarization

One form of scalarizing the regularized approximation problem (5.8) that is often considered in practice is to use the *relative weights*, *i.e.*,

$$\text{minimize} \quad \phi_1(f(x)-b)+\gamma\phi_2(x), \tag{5.10}$$

where $\gamma > 0$ represents the relative importance of the second objective compared to the first objective. According to this formulation, the first term $\phi_1(f(x)-b)$ in the objective is sometimes called the *primary objective*, and $\phi_2(x)$ is called the *regularization* term. When $0<\gamma<1$, we are more concerned about finding a good approximation of the system of equations $f(x)=b$, while when $\gamma>1$, we put more emphasis on finding a small vector x.

It is easily seen that the problem (5.10) can be obtained from (5.9) by dividing the objective by $\lambda_1>0$ and letting $\gamma=\lambda_2/\lambda_1$, and similarly, as γ varies over $\mathbf{R}_{++}$, the solution of (5.10) traces out the Pareto front of the original biobjective problem

(5.8). The advantage of using the relative scalarization formulation (5.10) is that, now we only need to choose a scalar hyperparameter $\gamma > 0$ instead of a vector in $\mathbf{R}^2_{++}$ to control the trade-off between the two objectives.

The regularization term $\phi_2(x)$ in (5.10) can be interpreted from different perspectives. In the context of model fitting, it represents our prior knowledge about some model parameter x, *e.g.*, it should not be too large. In this case, the factor γ can be interpreted as the strength of this prior knowledge, so a large γ means that we have a strong belief that the model parameter x should be small, while a small γ means that we are not very sure about this prior. From a modeling perspective, when the function f has the form $f(x) = Ax$, it might be a linear approximation of some more complicated model (say) $g(x)$, where the approximation $Ax \approx g(x)$ holds only when the vector x is small. Now the target system of equations $Ax = b$ is an approximation of the system of equations $g(x) = b$ that we actually want to solve, so the regularization term $\phi_2(x)$ can be interpreted as a penalty that encourages us to find a solution x that is small enough to make the linear approximation valid, and therefore this solution make sense. We can also interpret the regularization term from a Bayesian perspective; see §5.4.

5.2.2 Tikhonov regularization

The most common regularized approximation problem has the form

$$\text{minimize} \quad (\phi_1(f(x) - b),\ \|x\|_2^2) \tag{5.11}$$

where the regularization term

$$\phi_2(x) = \|x\|_2^2 = x_1^2 + \cdots + x_n^2$$

is the squared Euclidean norm of x, and is called the *Tikhonov regularization*. This regularization corresponds to applying a quadratic penalty $\psi(u) = u^2$ to each component of the variable x, and therefore encourages all components of x to be small (see §4.1.3 and figure 4.3). In particular, the penalty value assigned to some entry of x by a Tikhonov regularization grows very quickly as the magnitude of this entry increases, so it is very effective in preventing large entries appearing in the solution of (5.11).

Example 5.2 *Tikhonov regularization least squares.* If the primary objective $\phi_1(f(x) - b)$ in the problem (5.11) is the least squares cost, *i.e.*, in the form

$$\phi_1(f(x) - b) = \|Ax - b\|_2^2,$$

then the resulting problem

$$\text{minimize} \quad (\|Ax - b\|_2^2,\ \|x\|_2^2) \tag{5.12}$$

is called the *Tikhonov regularization least squares* problem, or *ridge regression*. This problem is a convex biobjective optimization problem, since both objectives $\|Ax - b\|_2^2$ and $\|x\|_2^2$ are convex functions of x.

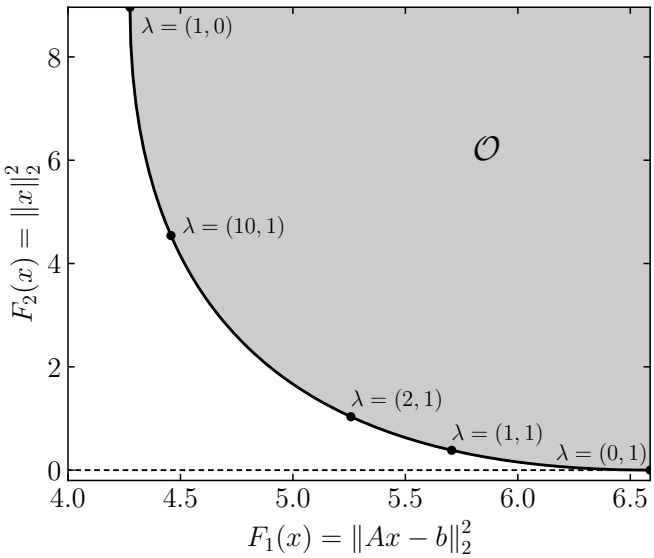

Figure 5.9 The set $\mathcal{O}$ of achievable objective values $(\|Ax - b\|_2^2, \|x\|_2^2)$ of the biobjective optimization problem (5.12) is shown shaded. The Pareto front, which is the lower left boundary of $\mathcal{O}$, is shown thick. Each dot on the Pareto front corresponds to a Pareto optimal value obtained from scalarization with some weight vector $\lambda \succ 0$.

We can scalarize the problem (5.12) with $\lambda \succ 0$ as

$$\begin{aligned} \lambda_1\|Ax - b\|_2^2 + \lambda_2\|x\|_2^2 &= \lambda_1(x^T A^T Ax - 2b^T Ax + b^T b) + \lambda_2 x^T x \\ &= x^T(\lambda_1 A^T A + \lambda_2 I)x - 2\lambda_1 b^T Ax + \lambda_1 b^T b. \end{aligned}$$

Since the matrix $\lambda_1 A^T A + \lambda_2 I \succ 0$ for all $\lambda \succ 0$, this scalarized objective is a convex quadratic function of x, which therefore achieves its minimum at

$$x = (\lambda_1 A^T A + \lambda_2 I)^{-1}\lambda_1 A^T b = (A^T A + \gamma I)^{-1} A^T b, \tag{5.13}$$

where $\gamma = \lambda_2/\lambda_1$ is the relative weight of the two objectives. From (5.13) we may see that after adding the Tikhonov regularization term $\|x\|_2^2$ to the least squares objective $\|Ax - b\|_2^2$, the solution requires no rank assumption on the data matrix A.

For each $\gamma > 0$, the point x given by (5.13) is a Pareto optimal point of the original biobjective optimization problem (5.12). By varying the parameter $\gamma \in \mathbf{R}_{++}$, we can obtain all different Pareto optimal points of the problem (5.12), except two extreme points that correspond to the limits $\gamma \to 0$ and $\gamma \to \infty$. In the first case, this Pareto optimal point can be obtained with the scalarization weight $\lambda = (1, 0)$ (which is indeed when $\gamma = 0$), and is the least squares (approximate) solution of the linear equation $Ax = b$, given by $x = (A^T A)^{-1} A^T b$ (assuming $m \geq n$ and A has full rank). In the second case, this Pareto optimal point corresponds to the scalarization weight $\lambda = (0, 1)$, and therefore is simply the zero vector $x = 0$.

Results of an example of the Tikhonov regularization least squares problem (5.12) with $m = 100$ and $n = 10$ is shown in figure 5.9, where the problem data $A \in \mathbf{R}^{100\times 10}$

and $b \in \mathbf{R}^{100}$ are generated randomly. For simplicity of notation, in the following discussion we may denote the two objectives as $F_1(x) = \|Ax - b\|_2^2$ and $F_2(x) = \|x\|_2^2$, so the problem (5.12) is in the form of the biobjective optimization problem (5.1) with $k = 2$. The set $\mathcal{O}$ of achievable objective values $(\|Ax - b\|_2^2, \|x\|_2^2)$ of the problem (5.12) is shown shaded, and the Pareto front (the lower left boundary of $\mathcal{O}$) is shown thick. Some examples of Pareto optimal values obtained from different λ are shown as dots on the Pareto front, and, in particular, the two extreme points mentioned above are the two endpoints.

From figure 5.9, we can say a lot about the trade-offs between the two objectives $F_1(x) = \|Ax - b\|_2^2$ and $F_2(x) = \|x\|_2^2$ in this example, for instance:

- The end point of the Pareto front on the left (*i.e.*, obtained with $\lambda = (1, 0)$) shows the best possible value of F_1 that can be achieved, without considering the other objective F_2 at all. At this Pareto optimal point x, the objective value of F_2 is the squared Euclidean norm of the least squares approximate solution of the linear equation $Ax = b$, *i.e.*, $\|(A^TA)^{-1}A^Tb\|_2^2$.
- The end point of the Pareto front on the right shows the best possible value of F_2 that can be achieved (which is exactly zero), without considering F_1. At this Pareto optimal point, in particular, the objective value of F_1 equals to $\|b\|_2^2$.
- As we move from the left to the right along the Pareto front, *i.e.*, as $\gamma \to \infty$, we are trading-off better performance in the objective F_2 for worse performance in the objective F_1. In particular, any vertical line $F_1(y) = \alpha$ that intersects with the Pareto front at some point $(\alpha, F_2(y))$ gives us the best possible value of F_2 that can be achieved when requiring F_1 to be at most α, and any horizontal line $F_2(y) = \beta$ that intersects with the Pareto front at some point $(F_1(y), \beta)$ gives us the best possible value of F_1 that can be achieved when requiring F_2 to be at most β.

5.2.3 Sparsity regularization

Another type of useful regularization that appear frequently in approximation problems in the *sparsity regularization*. In many applications, we may expect that the solution x of some approximation problem has only a few nonzero entries, *i.e.*, the solution x is *sparse*. For this purpose, we may consider the following regularized approximation problem:

$$\text{minimize} \quad (\phi_1(f(x) - b),\ \mathbf{card}\, x) \tag{5.14}$$

with variable $x \in \mathbf{R}^n$, where the regularization term

$$\phi_2(x) = \mathbf{card}\, x = |\{i \mid x_i \neq 0,\ i = 1, \ldots, n\}|$$

is the *cardinality function* that counts the number of nonzero entries in x. The cardinality function is nonconvex, and the problem (5.14) is in general a very hard combinatorial optimization problem.

Notice that $\mathbf{card}\, x \in \{0, 1, \ldots, n\}$ for all $x \in \mathbf{R}^n$. Hence, when n is small, one straightforward approach to find the Pareto front of the problem (5.14) is to check all possible sparsity patterns of x with $k \leq n$ nonzero entries. To illustrate this idea,

consider an example where the function f is linear given by $f(x) = Ax$ for some matrix $A \in \mathbf{R}^{m\times n}$. For each fixed sparsity pattern of x with k nonzero entries, we can find the best possible value of the first objective $\phi_1(Ax - b)$ by solving the problem

$$\text{minimize} \quad \phi_1(\tilde{A}\tilde{x} - b)$$

with variable $\tilde{x} \in \mathbf{R}^k$, where the matrix $\tilde{A} \in \mathbf{R}^{m\times k}$ consists of the columns of A that correspond to the k nonzero entries of x. Then to find the Pareto optimal point at $\mathbf{card}\, x = k$, this procedure is done for each of the $n!/(k!(n-k)!)$ possible sparsity patterns of x with k nonzero entries, and the best value of $\phi_1(Ax - b)$ among all these combinations is taken.

The ℓ_1-norm heuristic

The number of possible sparsity patterns grows exponentially with n, so directly solving (5.14) according to the procedure above is not computationally tractable when n is large (or even at moderate values). In this case, a good heuristic exists for finding an approximation of the Pareto front of the problem (5.14), which is to replace the cardinality function $\mathbf{card}\, x$ in (5.14) with the ℓ_1-*norm regularization*, *i.e.*, to solve the problem

$$\text{minimize} \quad (\phi_1(f(x) - b),\ \|x\|_1) \tag{5.15}$$

where the regularization term now becomes

$$\phi_2(x) = \|x\|_1 = |x_1| + \cdots + |x_n|.$$

Recall that the ℓ_1-norm is the convex envelope of the cardinality function (strictly speaking, within some constrained domain; see example 2.11), so the problem (5.15) is expected to be a good convex approximation of the cardinality regularized problem (5.14), given that the primary objective $\phi_1(f(x) - b)$ is convex.

The relationship between the two problems (5.14) and (5.15) can also be interpreted from a penalty function approximation perspective. The ℓ_1-norm regularization consists in applying an absolute value penalty $\psi(u) = |u|$ to each component of x, which adds (relatively) large penalty even to small values of the components (see figure 4.3, page 116). Hence, sparsity is encouraged in the solution of (5.15), similar to what the cardinality regularization does in the original cardinality regularized problem (5.14).

Assuming that the primary objective $\phi_1(f(x)-b)$ is convex, then scalarizing the problem (5.15) with $\gamma > 0$ gives us the convex program

$$\text{minimize} \quad \phi_1(f(x) - b) + \gamma\|x\|_1 \tag{5.16}$$

with variable $x \in \mathbf{R}^n$. When the primary objective $\phi_1(f(x) - b)$ is a least squares cost, *i.e.*, in the form $\|Ax - b\|_2^2$, the problem (5.16) is sometimes called the *lasso regression*. Noticing that taking a larger value of the hyperparameter γ in (5.16) leads to a sparser solution of this problem, a solution of (5.16) with the smallest $\gamma > 0$ that has cardinality k hence serves as a good approximation of the Pareto optimal point of (5.14) at $\mathbf{card}\, x = k$. Such an approximation via the ℓ_1-norm

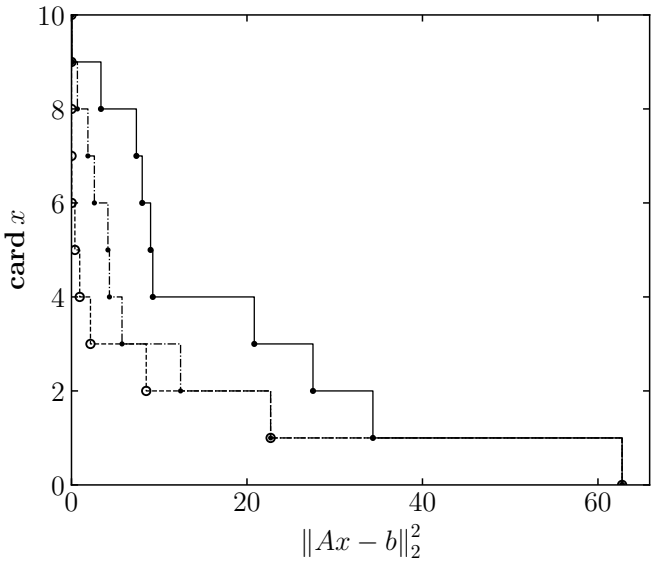

Figure 5.10 Optimal trade-off curve of the sparsity regularization least squares problem (5.18) with $m = 10$ and $n = 20$. The circles on the dashed curve are the (globally) Pareto optimal values of the problem, while the dots on the solid and dashdotted curve are approximations obtained from the ℓ_1-norm heuristic (5.16) and polishing (5.17), respectively.

heuristic can be further *polished* by fixing its sparsity pattern and find the value of x that minimizes $\phi_1(f(x) - b)$, *i.e.*, by solving the problem

$$\begin{array}{ll} \text{minimize} & \phi_1(f(x) - b) \\ \text{subject to} & x_i = 0, \quad i \in \mathcal{I}_0, \end{array} \tag{5.17}$$

where the set $\mathcal{I}_0$ consists of the $(n - k)$ zero indexes corresponding to the solution of (5.16).

Example 5.3 *Sparsity regularization least squares.* We consider a regularized approximation problem in the form (5.14), applied to the system of linear equations $Ax = b$, which is given by the biobjective optimization problem

$$\text{minimize} \quad (\|Ax - b\|_2^2, \ \mathbf{card}\, x), \tag{5.18}$$

where $x \in \mathbf{R}^n$ is the variable and $A \in \mathbf{R}^{m\times n}$, $b \in \mathbf{R}^m$ are given data. The problem (5.18) is called a *sparsity regularization least squares* problem, or a *regressor selection* problem.

Figure 5.10 shows a numerical example of the problem (5.18) with $m = 10$ and $n = 20$, where the problem data $A \in \mathbf{R}^{10\times 20}$, $b \in \mathbf{R}^{10}$ are generated randomly. The circles on the dashed step lines are the (globally) Pareto optimal values from directly solving the problem (5.18) via the combinatorial optimization procedure described above. The dots on the solid curve are approximations obtained from the ℓ_1-norm heuristic via

(5.16) with primary objective $\|Ax - b\|_2^2$, and the dots on the dashdotted curve are the further polished approximations from (5.17).

We have several observations from the figure. First of all, the point with vertical axis value $\mathbf{card}\, x = 0$ corresponds to the Pareto optimal value $(\|b\|_2^2, 0)$ of the problem (5.18). Moreover, when $\mathbf{card}\, x = 1$, the heuristic method based on the ℓ_1-norm regularization and polishing actually finds the true global Pareto optimum. For the other cases, the approximations obtained from this heuristic are also quite close to the ground truth.

5.3 Smoothing

Another class of problems that are closely related to regularized approximations are the *smoothing* problems. Consider a one-dimensional time series signal $x \in \mathbf{R}^n$ that is unknown, where each entry x_i for $i = 1, \ldots, n$ represents the value of some function of time, evaluated at the ith time step. The goal of smoothing is to find a good estimate $\hat{x}$ of the true signal x, based on some *noisy* (or *corrupted*) observation $x_{\text{obs}} \in \mathbf{R}^n$ of x, which is assumed to be generated according to

$$x_{\text{obs}} = x + v,$$

where $v \in \mathbf{R}^n$ is some noise or corruption vector. Additionally, we have the prior knowledge that the true signal is (relatively) large and somehow *smooth*, *i.e.*, it does not change too much from one time step to the next. In other words, the relationship

$$x_i \approx x_{i+1}$$

holds for most $i = 1, \ldots, n-1$. The unknown noise v is assumed to be, on the contrary, small and rapidly changing.

There are several other names for the smoothing problems, for example, since we are trying to recover the true signal x from noisy observations, they are also called *reconstruction* problems. It is also called *denoising* problems, since the goal is to remove the noise from the observed signal x_{obs} to recover the true signal x. These ideas can be readily extended to higher-dimensional time series, for example, multichannel signals, video, images, etc.

5.3.1 Problem formulation

In the most general case, smoothing problems can be formulated as regularized approximation problems in the form (5.8), which is given by

$$\text{minimize} \quad (\|\hat{x} - x_{\text{obs}}\|_2^2,\ \phi(\hat{x})), \tag{5.19}$$

where $\hat{x} \in \mathbf{R}^n$ is the variable representing the estimated signal, and the function $\phi\colon \mathbf{R}^n \to \mathbf{R}$ controls the smoothness of the estimation. Therefore, by solving the problem (5.19), we aim at finding an estimation $\hat{x}$ that is close to the observed signal x_{obs} in the sense of the least squares cost $\|\hat{x} - x_{\text{obs}}\|_2^2$, while also being smooth in the sense of the regularization term $\phi(\hat{x})$.

We should note that, strictly speaking, the regularization terms in the problems (5.8) and (5.19) serve slightly different purposes. Here in the problem (5.19), the regularization term $\phi(\hat{x})$ is designed to encourage the estimated signal $\hat{x}$ to be *smooth*, while in the problem (5.8), the regularization term $\phi_2(x)$ is typically chosen to encourage the variable x to be *small*.

There are several obvious observations about the problem (5.19). First of all, one Pareto optimal point of this problem is simply $\hat{x} = x_{\text{obs}}$, which corresponds to the case where we do not consider the smoothness of the estimation at all, *i.e.*, simply taking the observed signal as the estimation. Another extreme Pareto optimal point is the one that minimizes the smoothness regularization term $\phi(\hat{x})$ without considering the approximation or recovery error $\|\hat{x} - x_{\text{obs}}\|_2^2$ at all. In many cases (see exercise 5.3 for an exception), this Pareto optimal point corresponds to a constant signal $\hat{x} = \hat{c}\mathbf{1}$ for some constant $\hat{c} \in \mathbf{R}$, and the value of $\hat{c}$ is usually the average of the entries of the observed signal x_{obs}, *i.e.*,

$$\hat{c} = \operatorname*{argmin}_{c \in \mathbf{R}} \|x_{\text{obs}} - c\|_2^2 = (1/n)\mathbf{1}^T x_{\text{obs}},$$

which minimizes the least squares cost $\|\hat{x} - x_{\text{obs}}\|_2^2$ among all constant signals.

To trade-off between the signal recovery objective $\|\hat{x} - x_{\text{obs}}\|_2^2$ and the smoothness objective $\phi(\hat{x})$, we can scalarize the problem (5.19) with $\gamma > 0$ as

$$\text{minimize} \quad \|\hat{x} - x_{\text{obs}}\|_2^2 + \gamma\phi(\hat{x}), \tag{5.20}$$

where the hyperparameter γ controls the relative importance of the two objectives in (5.19). A small value of γ leads to an estimation $\hat{x}$ that is close to the observed signal x_{obs}, while a large value of γ leads to a very smooth estimation $\hat{x}$. In particular, the two extreme Pareto optimal points mentioned above correspond to the limits $\gamma \to 0$ and $\gamma \to \infty$, respectively.

The function ϕ can be designed in different ways to meet various application needs, but it is generally chosen to be a convex function of $\hat{x}$ that penalizes rapid changes in the estimated signal, so that the resulting problem (5.19) (and hence (5.20)) is convex. We introduce some common choices of the function ϕ in the following sections.

5.3.2 Quadratic smoothing

The most common choice of the smoothness regularization function ϕ in the problem (5.19) is the *quadratic smoothing regularization*, which is given by

$$\phi(\hat{x}) = \sum_{i=1}^{n-1} (\hat{x}_{i+1} - \hat{x}_i)^2. \tag{5.21}$$

This regularization penalizes the first-order difference $\hat{x}_{i+1} - \hat{x}_i$ quadratically at each time step of the estimated signal $\hat{x}$, which encourages the estimation to be smooth in the sense that it does not change too much from one time step to the next. The quadratic smoothing regularization (5.21) can also be represented in

a more compact form as $\phi(\hat{x}) = \|D\hat{x}\|_2^2$, where the matrix $D \in \mathbf{R}^{(n-1)\times n}$ is the bidiagonal first-order difference matrix defined as

$$D = \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & -1 & 1 \end{bmatrix}. \tag{5.22}$$

The objective of the corresponding scalarized smoothing problem (5.20) with the quadratic smoothing regularization (5.21) is given by

$$\begin{aligned} \|\hat{x} - x_{\text{obs}}\|_2^2 + \gamma\|D\hat{x}\|_2^2 &= (\hat{x} - x_{\text{obs}})^T(\hat{x} - x_{\text{obs}}) + \gamma\hat{x}^T D^T D\hat{x} \\ &= \hat{x}^T(I + \gamma D^T D)\hat{x} - 2x_{\text{obs}}^T\hat{x} + x_{\text{obs}}^T x_{\text{obs}}, \end{aligned}$$

which is a convex quadratic function of $\hat{x}$, and therefore achieves its minimum at

$$\hat{x} = (I + \gamma D^T D)^{-1} x_{\text{obs}}.$$

Example 5.4 *Quadratic smoothing.* Figure 5.11 shows an example of quadratic smoothing. The time series on the top panel is the true signal $x \in \mathbf{R}^{5000}$ generated randomly, and the time series on the second row is the observed signal x_{obs}. It is seen that the observed signal x_{obs} has a lot of rapidly changing noise.

To obtain a smooth estimation $\hat{x}$ of the true signal x based on the observation x_{obs}, we solve the scalarized quadratic smoothing problem

$$\text{minimize} \quad \|\hat{x} - x_{\text{obs}}\|_2^2 + \gamma\|D\hat{x}\|_2^2 \tag{5.23}$$

with variable $\hat{x} \in \mathbf{R}^{5000}$ for different values of the hyperparameter γ, where the first-order difference matrix $D \in \mathbf{R}^{4999\times 5000}$ is given by (5.22). Three examples of the resulting estimation $\hat{x}$ are shown in the last three rows of figure 5.11, where the value of γ decreases from top to bottom. It is observed in the figure that when $\gamma = 0.5$, the estimation $\hat{x}$ is smoother than x_{obs}, but it still includes a lot of noise. Then when we increase γ to 100, the estimation $\hat{x}$ becomes much smoother, and is very close to the true signal x. If we continue to increase the value of γ to 10000, on the one hand, the estimation $\hat{x}$ becomes even smoother, but on the other hand, it becomes too 'flat' and many characteristics of the true signal x are lost. Based on these observations, we may expect that the estimation $\hat{x}$ corresponding to $\gamma \to \infty$ will eventually converge to the constant signal $\hat{x} = 0$.

Figure 5.12 shows (a part of) the Pareto front of the original biobjective smoothing problem (5.19) with the quadratic smoothing regularization (5.21), where the three dots on the Pareto front corresponds to the Pareto optimal values obtained from scalarization with $\gamma = 0.5, 100, 10000$. Notice that when $\gamma < 100$, a small decrease in the approximation error $\|\hat{x} - x_{\text{obs}}\|_2^2$ is associated with a large increase in the smoothness regularization $\|D\hat{x}\|_2^2$. This indicates that in this region of the Pareto front, there is a strong trade-off between the primary signal recovery objective $\|\hat{x} - x_{\text{obs}}\|_2^2$ and the

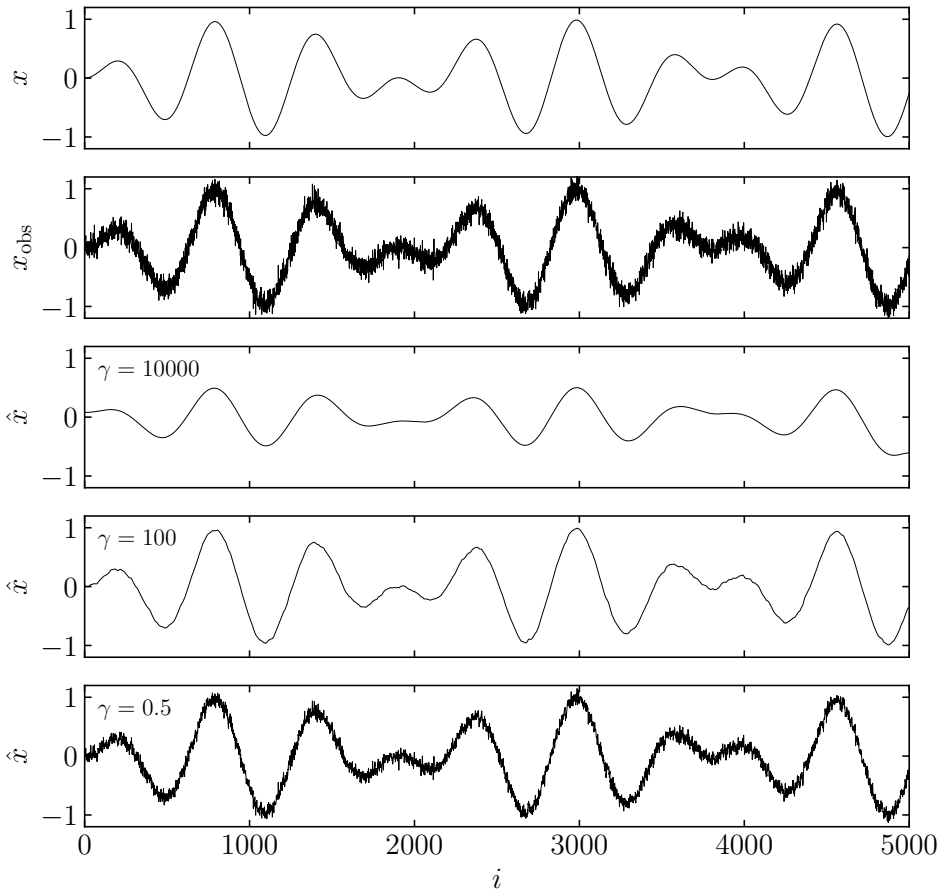

Figure 5.11 *Quadratic smoothing.* The first and second rows show the true signal $x \in \mathbf{R}^{5000}$ and the observed signal x_{obs}, respectively. The last three rows show the estimation $\hat{x}$ obtained from solving the quadratic smoothing problem (5.23) with different values of the hyperparameter γ.

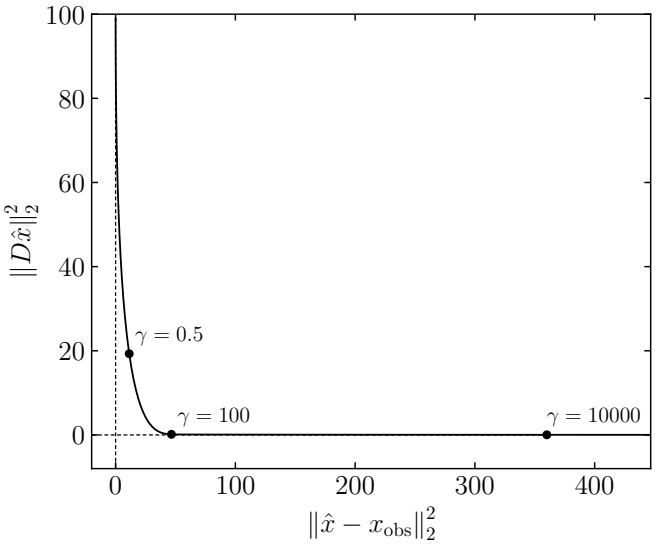

Figure 5.12 Optimal trade-off curve corresponds to the (original, unscalarized) quadratic smoothing problem (5.23). The three dots are the Pareto optimal values obtained from solving (5.23) with $\gamma = 0.5$, 100, and 10000, respectively.

smoothness regularization objective $\|D\hat{x}\|_2^2$. In other words, we can achieve a significant improvement in the smoothness of the estimation $\hat{x}$ with a small sacrifice in the recovery performance. However, when $\gamma > 100$, continuing to increase its value does not further improve the smoothness of the estimation $\hat{x}$ too much (which is indeed the case since the value of $\|D\hat{x}\|_2^2$ is already very close to zero when $\gamma = 100$), while the approximation error $\|\hat{x} - x_{\text{obs}}\|_2^2$ increases significantly. This suggests that in this region of the Pareto front, the trade-off between these two objectives are weak, *i.e.*, we need to sacrifice a lot of recovery performance to achieve only a minor improvement in the smoothness of the estimation $\hat{x}$, and is hence not desirable. Therefore, the Pareto optimal point corresponding to $\gamma = 100$ is sometimes called the 'knee point' of the Pareto front (separating the strong and weak trade-off regions), which is usually a good choice for the hyperparameter γ in practice.

5.3.3 Total variation smoothing

The idea of quadratic smoothing works quite well when the true signal x is relatively smooth (as in figure 5.11), but it may not be suitable when the true signal x has some abrupt changes, for example, a square wave signal that is piecewise constant with some jumps. In these cases, the quadratic smoothing regularization (5.21) may also remove these rapid variations in the signal, which can actually be very important characteristics of the true signal x.

To provide a smooth estimation $\hat{x}$ that can still capture some abrupt changes in the true signal x, we can consider the *total variation smoothing*, which is given by

the regularization function

$$\phi(\hat{x}) = \sum_{i=1}^{n-1} |\hat{x}_{i+1} - \hat{x}_i| = \|D\hat{x}\|_1, \tag{5.24}$$

where the matrix $D \in \mathbf{R}^{(n-1)\times n}$ is the same first-order difference matrix defined in (5.22). The total variation smoothing regularization (5.24) adds an absolute value penalty to the first-order difference $\hat{x}_{i+1} - \hat{x}_i$ at each time step of the estimated signal $\hat{x}$, which encourages the estimation to be smooth in the sense that the total variations in the signal is *sparse*, *i.e.*, the signal should be roughly piecewise constant with only a few jumps.

Example 5.5 *Total variation smoothing.* To illustrate the properties of total variation smoothing, and to compare it with the quadratic smoothing, we consider the time series shown in figure 5.13. The true signal x shown in the top panel is approximately a square wave signal that is roughly piecewise constant with several large jumps. The observed noisy signal x_{obs} shown in the second row is generated by adding some rapidly changing noise (with smaller amplitude) to the true signal x. It is obvious that except for the noise, there are several rapid variations in x_{obs} inherited from the true signal x, and we would like to capture these important characteristics in the estimation $\hat{x}$.

We first apply the total variation smoothing to x_{obs} to obtain a smooth estimation $\hat{x}$ of the true signal x, by solving the (scalarized) problem

$$\text{minimize} \quad \|\hat{x} - x_{\text{obs}}\|_2^2 + \gamma\|D\hat{x}\|_1 \tag{5.25}$$

with variable $x \in \mathbf{R}^{5000}$, where the first-order difference matrix $D \in \mathbf{R}^{4999\times 5000}$ is given by (5.22). The resulting estimation $\hat{x}$ for different values of the hyperparameter γ are shown in the last three rows of figure 5.13. When taking $\gamma = 0.1$, the estimation $\hat{x}$ is quite close to the observed signal x_{obs}, and it still includes a lot of noise. When we increase γ to 1, most of the noise is removed from the estimation $\hat{x}$, and the resulting estimation $\hat{x}$ is very close to the true signal x. When we continue to increase γ to 100, the jumps in the estimation $\hat{x}$ are even sparser, and the estimation $\hat{x}$ becomes more like a piecewise constant signal. In this case, on the one hand, the estimation $\hat{x}$ is indeed very smooth in the sense that it has only a few jumps, but on the other hand, it is not so close to the true signal x as the estimation obtained with $\gamma = 1$. Nevertheless, in all three cases, the major jumps in the true signal x are well captured by the estimation $\hat{x}$.

Figure 5.14 shows a part of the optimal trade-off curve of this total variation smoothing example obtained from taking different values of the hyperparameter γ in (5.25). The three dots correspond to $\gamma = 0.1$, 1, and 100, respectively. The 'knee point' of this Pareto front is clearly around the Pareto optimal value with $\gamma = 1$, where continuing to increase γ does not further improve the smoothness of the estimation $\hat{x}$ too much, but the approximation error $\|\hat{x} - x_{\text{obs}}\|_2^2$ increases significantly.

Now we apply the quadratic smoothing to the same observed signal x_{obs} in figure 5.13 by solving the problem (5.23) with different values of γ. The resulting estimation $\hat{x}$ is shown in figure 5.15. When taking $\gamma = 1$ in the problem (5.23), the estimation $\hat{x}$ from the quadratic smoothing is more or less similar to the one from the total variation smoothing shown on the bottom of figure 5.13, which is quite close to the observed

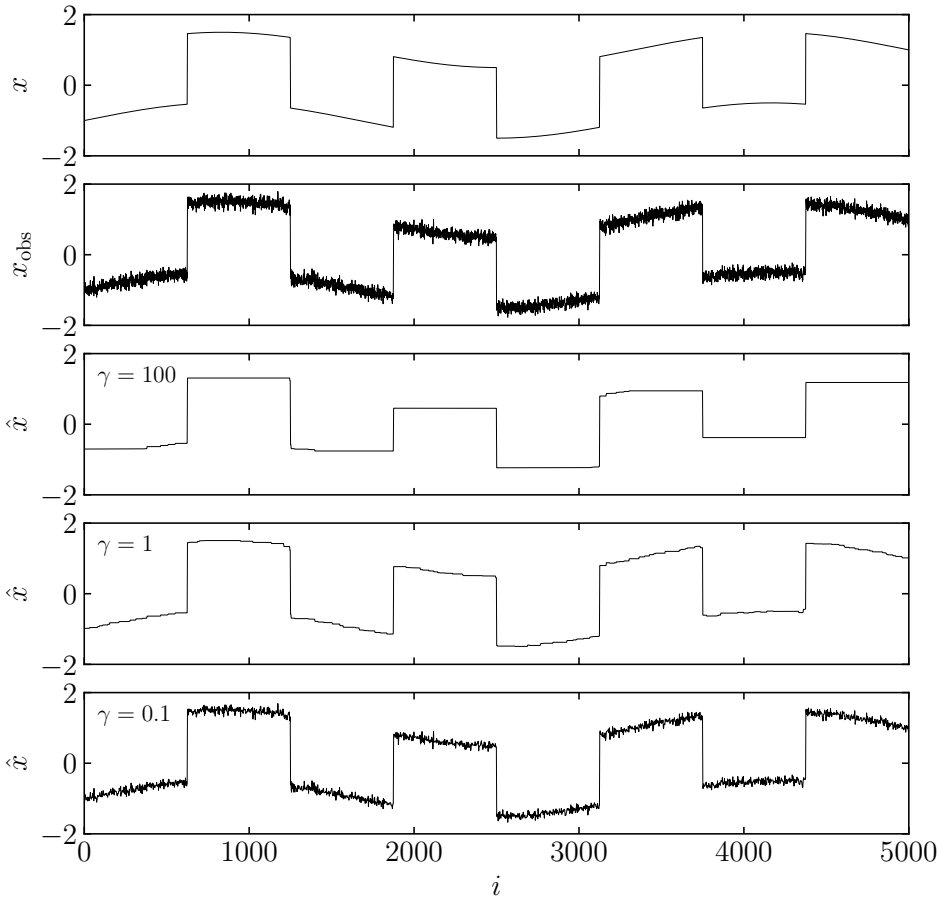

Figure 5.13 *Total variation smoothing.* The first and second rows show the true signal $x \in \mathbf{R}^{5000}$ and the observed signal x_{obs}, respectively. The last three rows show the estimation $\hat{x}$ obtained from solving the total variation smoothing problem (5.25) with different values of the hyperparameter γ.

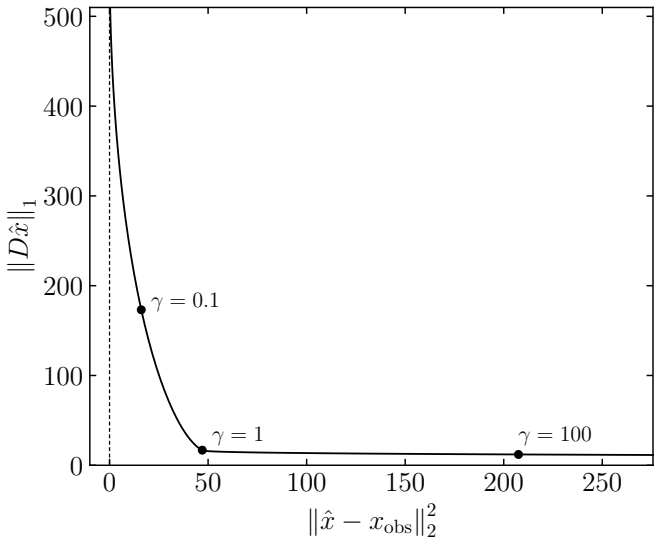

Figure 5.14 Optimal trade-off curve of the total variation smoothing example from solving (5.25) with different values of the hyperparameter γ, where the three dots correspond to $\gamma = 0.1$, 1, and 100, respectively.

signal x_{obs} and still includes a lot of noise. Then as we increase the value of γ, the estimation $\hat{x}$ from the quadratic smoothing does become smoother in the sense that the noise is reduced, but the sharp jumps in the true signal x are also smoothed out, as shown in the first three rows of figure 5.15. In particular, when $\gamma = 10000$, the estimation $\hat{x}$ is more close to a sinusoidal signal rather than a piecewise constant signal. From this example we may see that the quadratic smoothing is not suitable when the true signal is known to have rapid variations, since in this case, no matter how we choose the hyperparameter γ in the problem (5.23), the resulting estimation $\hat{x}$ from the quadratic smoothing is either keeps still a lot of noise, or becomes too smooth and loses the important jumps in the true signal x.

5.4 Maximum a posteriori estimation

The idea of regularization can be applied to the maximum likelihood estimation framework presented in §4.2, which leads to the class of *maximum a posteriori* (MAP) estimation problems.

5.4.1 Problem formulation

Let $x \in \mathbf{R}^n$ be some variable that we want to estimate based on the observed data $y \in \mathbf{R}^m$, and assume that both x and y are random variables with joint probability distribution $p\colon \mathbf{R}^n \times \mathbf{R}^m \to \mathbf{R}_+$, where the value $p(x, y)$ represents the corresponding density. Note the difference here from the maximum likelihood

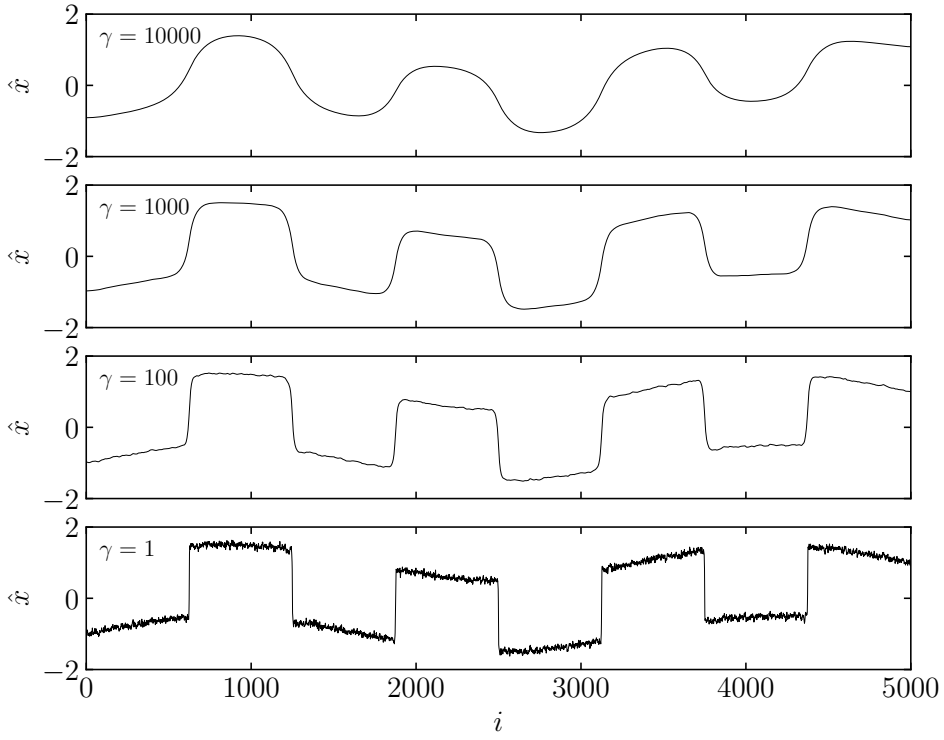

Figure 5.15 Quadratic smoothing applied to the same signal as in figure 5.13. Each row corresponds to an estimation $\hat{x}$ obtained from solving the problem (5.23) with different values of the hyperparameter γ.

estimation framework, where only the observation y is considered as a random variable, and the variable x is treated as a deterministic parameter to be estimated.

The general idea of MAP estimation is to find a value of x that maximizes the *posterior distribution* of x given the observation y, defined as $p_{x|y}\colon \mathbf{R}^n \times \mathbf{R}^m \to \mathbf{R}_+$ with density $p_{x|y}(x, y)$, which represents our knowledge about the variable x after observing y. Let

$$p_x(x) = \int p(x, y)\ dy$$

and

$$p_y(y) = \int p(x, y)\ dx$$

be the marginal probability density functions of x and y, respectively, where the *prior distribution* $p_x\colon \mathbf{R}^n \to \mathbf{R}_+$ can be interpreted as our prior knowledge about the random variable x before observing any data y, and $p_y\colon \mathbf{R}^m \to \mathbf{R}_+$ can be interpreted as the prior information about the observation y with the current chosen model (under all possible values of its parameter x). According to the Bayes rule, the conditional probability distribution of y given x is then given by the function $p_{y|x}\colon \mathbf{R}^n \times \mathbf{R}^m \to \mathbf{R}_+$ where

$$p_{y|x}(x, y) = \frac{p(x, y)}{p_x(x)},$$

which takes the role of the likelihood function of the variable x in the maximum likelihood estimation setup. Hence, the posterior distribution $p_{x|y}$ can be expressed in terms of these probability distributions as

$$p_{x|y}(x, y) = \frac{p(x, y)}{p_y(y)} = \frac{p_{y|x}(x, y)p_x(x)}{p_y(y)}. \tag{5.26}$$

Noticing that for each fixed value of the observation y, the denominator $p_y(y)$ in (5.26) is a constant which is not influenced by the value of the variable x, the MAP estimation problem can be therefore formulated as

$$\text{maximize} \quad p_{y|x}(x, y)p_x(x), \tag{5.27}$$

where the variable is $x \in \mathbf{R}^n$, and the problem data is the observation $y \in \mathbf{R}^m$. Again, it is generally more convenient to work with the logarithm of the posterior distribution, so we have the equivalent formulation given by

$$\text{maximize} \quad \log p_{y|x}(x, y) + \log p_x(x). \tag{5.28}$$

Now it is easily seen that if the logarithm of the posterior distribution $p_{y|x}$ is concave in x for each fixed y, and the logarithm of the prior distribution p_x is also concave, then the MAP estimation problem (5.28) is a convex (in particular, concave maximization) problem.

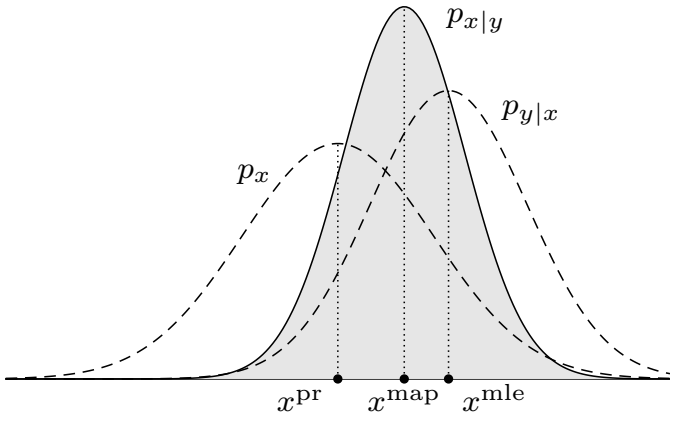

Figure 5.16 *Bayesian interpretation of MAP estimation.* The dashed curves represent the prior distribution p_x and the likelihood function $p_{y|x}$, respectively. The shaded area with solid boundary is the posterior distribution $p_{x|y}$ according to (5.26). The points x^{pr}, x^{mle}, and x^{map} achieve the maximum probability of the prior distribution p_x, the likelihood function $p_{y|x}$, and the posterior distribution $p_{x|y}$, respectively.

Bayesian interpretation

According to the problem formulation above, the MAP estimation problem (5.27) can be considered a Bayesian version of maximum likelihood estimation. Recall that a maximum likelihood estimation problem tries to find a value of x that maximizes the likelihood function $p_{y|x}(x, y)$ of x given some observation y. According to the Bayes rule in (5.26), the MAP estimation problem (5.27), on the other hand, tries to find a value of x that maximizes the posterior distribution $p_{x|y}(x, y)$ of x given y, *i.e.*, to find a value of x that not only leads to a high probability of observing y (which is captured by the likelihood term $p_{y|x}(x, y)$), but also has a high probability of being the true value of the variable according to our prior knowledge about x (which is captured by the term $p_x(x)$).

This interpretation is illustrated in figure 5.16. The dashed curves represent the prior distribution p_x of the variable x and the likelihood function $p_{y|x}$ of the variable x given the observation y, respectively. The prior distribution p_x achieves its maximum at the point x^{pr}, which is the value of x that is most likely to be the true value according to our prior knowledge. The likelihood function $p_{y|x}$ achieves its maximum at the point x^{mle}, which is the value of x that leads to the highest probability of observing y, *i.e.*, the maximum likelihood estimation of x. The shaded area with solid boundary represents the posterior distribution $p_{x|y}$ of x given y, *i.e.*, the (pointwise, for each x) product of the prior distribution p_x and the likelihood function $p_{y|x}$ (and normalized by the constant $p_y(y)$, as in (5.26)). The MAP estimation of x is the point x^{map} that achieves the maximum of the posterior distribution $p_{x|y}$, which is the value of x that corresponds to the best balance between being likely to be the true value according to our prior knowledge, and leading to a

high probability of observing y.

Regularization interpretation

If we compare the MAP estimation problem (5.28) with the maximum likelihood estimation problem ((4.20), page 124), we can see that the first term $\log p_{y|x}(x, y)$ here is essentially the same as the log-likelihood function in maximum likelihood estimation, and the only difference in the two objectives is the additional term $\log p_x(x)$ in the MAP estimation objective.

As a result, we can interpret the MAP estimation as a regularized version of the maximum likelihood estimation, where the first term $\log p_{y|x}(x, y)$ serves as the primary objective that tries to find the maximum likelihood estimation of x under which the probability of observing y has the highest value, while the second term $\log p_x(x)$ serves as a regularization term that incorporates our prior knowledge about the variable x into the estimation. Specifically, when some value of x leads to a small probability of the prior $p_x(x)$, then the term $\log p_x(x)$ will introduce a large negative penalty to the objective of the MAP estimation problem (5.28). In other words, if some value of x is less likely to happen according to our prior knowledge, then the MAP estimation problem (5.28) will be discouraged from taking this value of x as a solution.

5.4.2 Trade-off between likelihood and prior

In the MAP estimation setup, we also have to consider the trade-off between the log-likelihood term $\log p_{y|x}(x, y)$ and the prior term $\log p_x(x)$ in the objective of the problem (5.28), which is more or less similar to the trade-off between the approximation error and the regularization term in regularized approximations. The only difference here is that the trade-off parameter is usually implicitly determined by the shape of the prior distribution p_x, rather than being explicitly controlled by some hyperparameter as in the regularized approximation problems, *e.g.*, the scalarization weight γ in (5.10).

Roughly speaking, if the prior distribution p_x is 'flat', which means that it does not have a strong preference for any particular value of x, then the MAP estimation problem (5.28) solution will be more dominated by the log-likelihood term $\log p_{y|x}(x, y)$, since the prior term $\log p_x(x)$ only has a weak regularization effect to the objective. On the other hand, if the prior distribution p_x is very 'peaky', *i.e.*, it has a strong preference for some particular value of x, then the MAP estimation problem (5.28) will be more influenced by the prior term $\log p_x(x)$, and the resulting estimation of x will be more close to the value that is preferred by the prior distribution p_x, even if this value of x does not lead to the highest probability of observing y.

Example 5.6 *Trade-off in MAP estimation.* Consider a dataset (a_i, y_i), $i = 1, \ldots, m$, generated according to the linear model

$$y_i = a_i^T x + v_i,$$

where $x \in \mathbf{R}^n$ is the unknown parameter to be estimated, $a_i \in \mathbf{R}^n$ are known measurement vectors, and $v_i \in \mathbf{R}$ are the IID noise from standard Gaussian distribution

$\mathcal{N}(0,1)$. To formulate the MAP estimation problem (5.28), the log-likelihood function $\log p_{y|x}$ given this dataset and the noise model can be expressed as

$$\log p_{y|x}(x,y) = \sum_{i=1}^{m} \left(-\frac{\log(2\pi)}{2} - \frac{(y_i - a_i^T x)^2}{2} \right) = -\frac{m}{2}\log(2\pi) - \frac{1}{2}\|Ax-y\|_2^2,$$

where $A = \begin{bmatrix} a_1 & \cdots & a_m \end{bmatrix}^T \in \mathbf{R}^{m\times n}$ is the measurement matrix, and $y \in \mathbf{R}^m$ is the vector of observations (see §4.2.1, page 125). Assuming that the prior distribution p_x is a (multivariate) Gaussian $\mathcal{N}(0, \sigma_x^2 I)$ with mean zero and covariance matrix $\sigma_x^2 I$ (*i.e.*, the entries of x is assumed to be IID Gaussian variables with mean zero and variance σ_x^2), the log-prior term $\log p_x(x)$ can be similarly expressed as

$$\log p_x(x) = -\frac{n}{2}\log(2\pi\sigma_x^2) - \frac{1}{2\sigma_x^2}\|x\|_2^2.$$

Then the MAP estimation problem (5.28) has objective

$$\log p_{y|x}(x,y) + \log p_x(x) = -\frac{m}{2}\log(2\pi) - \frac{n}{2}\log(2\pi\sigma_x^2) - \frac{1}{2}\|Ax-y\|_2^2 - \frac{1}{2\sigma_x^2}\|x\|_2^2,$$

where the first two terms are constants that do not depend on x. Hence, the problem (5.28) can be equivalently formulated as

$$\text{minimize} \quad \|Ax-y\|_2^2 + (1/\sigma_x^2)\|x\|_2^2 \tag{5.29}$$

with variable $x \in \mathbf{R}^n$, where σ_x^2 is a hyperparameter that controls the variance of the Gaussian prior distribution p_x. This problem is essentially the same as the Tikhonov regularized least squares problem (5.12) in scalarized form. The only difference is that now the hyperparameter σ_x^2 has a clear statistical interpretation.

Figure 5.17 shows a numerical example of solving the problem (5.29) with $n = m = 100$ for different values of the hyperparameter σ_x^2, where the the problem data $A \in \mathbf{R}^{100\times 100}$ and $y \in \mathbf{R}^{100}$ are generated randomly. Let $\hat{x} \in \mathbf{R}^{100}$ be the solution of the MAP estimation problem (5.29) for a given value of σ_x^2, the histogram of the component amplitudes of $\hat{x}$ and the optimal residual $A\hat{x} - y$ are shown as shaded and empty bars, respectively. The solid and dashed curves in the figure are the (scaled) probability density functions of the Gaussian distribution $\mathcal{N}(0, \sigma_x^2)$ and $\mathcal{N}(0,1)$, which are the assumed prior distributions for each entry of x and $A\hat{x} - y$, respectively.

It is observed in figure 5.17 that when σ_x^2 takes a small value 0.25, the solution $\hat{x}$ has a small variance, and its component amplitudes are mostly distributed close to zero. This is consistent with the fact that the Gaussian prior distribution $\mathcal{N}(0, \sigma_x^2)$ with small variance has a strong preference for values of x close to zero. In this case, the amplitude distribution of the optimal residual $A\hat{x} - y$ is relatively widely spread out, compared to the amplitude distribution of $\hat{x}$. This also makes sense since the prior distribution $\mathcal{N}(0,1)$ for the residual $Ax - y$ has a larger variance than the prior distribution $\mathcal{N}(0, 0.25)$ for x. When $\sigma_x^2 = 1$, the solution $\hat{x}$ has a larger variance than the previous case, and since now the prior distribution for x has the same variance as the prior distribution for the residual $Ax - y$, the amplitude distributions of $\hat{x}$ and the optimal residual $A\hat{x} - y$ are more or less similar. If we continue to increase the value of σ_x^2 to 10, the prior distribution for x is almost flat, and therefore the amplitude distribution of $\hat{x}$ is much more widely spread out than the previous two cases. On the other hand, now the amplitude distribution of the optimal residual $A\hat{x} - y$ is mostly concentrated around zero. To sum up, by choosing a smaller value of σ_x^2, we

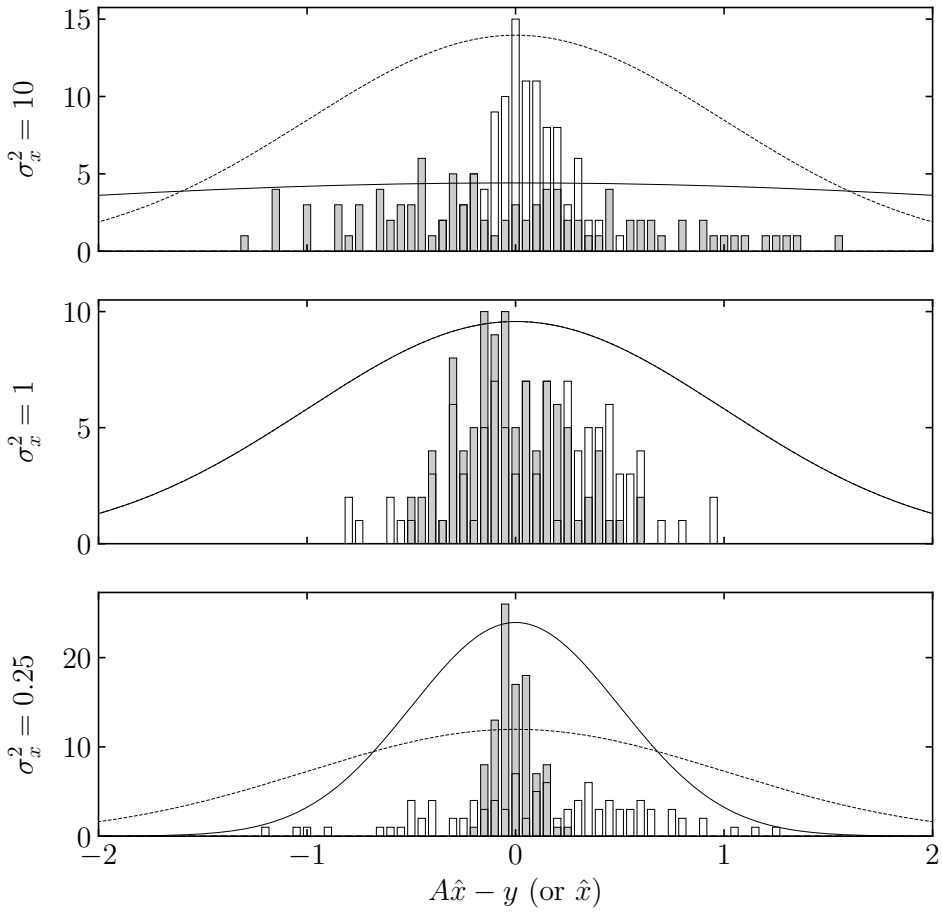

Figure 5.17 Histograms of the component amplitudes of the MAP estimation solution $\hat{x}$ (shaded bars) and the optimal residual $A\hat{x} - y$ (empty bars) obtained from solving the problem (5.29) with different values of the hyperparameter σ_x^2. The solid curve in each plot shows the (scaled) probability density functions of the Gaussian distribution $\mathcal{N}(0, \sigma_x^2)$. The standard Gaussian distribution $\mathcal{N}(0, 1)$ is also shown as the dashed curve for reference.

are putting a stronger belief on the prior knowledge about x that it should be close to zero, which means that we are trading-off a smaller estimation of $\hat{x}$ for a larger residual $A\hat{x} - y$.

These observations can also be interpreted from the perspective of regularized approximation, by regarding the term $\|Ax - y\|_2^2$ in the objective of the problem (5.29) simply as the primary approximation objective (as in, *e.g.*, the problem (5.12)), and the term $\|x\|_2^2$ as the regularization that incorporates our prior knowledge about x. Under this context, choosing a smaller value of σ_x^2 corresponds to putting a larger scalarization weight on the Tikhonov regularization term $\|x\|_2^2$, which therefore encourages the solution $\hat{x}$ to have a smaller norm. This is consistent with the previous observations.

5.4.3 Selection of the prior distribution

From the previous discussions, it is easily seen than all those maximum likelihood estimation problems presented in §4.2 can be readily extended to the MAP estimation framework (5.28) by introducing some kind of prior distribution regularization term $\log p_x(x)$ to the corresponding objectives, so that the desired characteristics according to our prior knowledge might be reflected in the estimation result of the variable x. In this section, we introduce several useful prior distributions that appear frequently in the MAP estimation problems.

As a simple reference point, in the following discussions we assume that the log-likelihood term $\log p_{y|x}(x, y)$ in the objective of problem (5.28) takes the form

$$\log p_{y|x}(x, y) = -\frac{m}{2}\log(2\pi) - \frac{1}{2}\|Ax - y\|_2^2, \tag{5.30}$$

where $x \in \mathbf{R}^n$ is the variable to be estimated, $A = \begin{bmatrix} a_1 & \cdots & a_m \end{bmatrix}^T \in \mathbf{R}^{m\times n}$ and $y \in \mathbf{R}^m$ are the given data, according to the linear measurement model $y_i = a_i^T x + v_i$ with IID standard Gaussian noise $v_i \sim \mathcal{N}(0, 1)$, as in the previous example. All results presented here are readily extended when the likelihood function $p_{y|x}$ has a different expression.

Gaussian prior

A Gaussian prior distribution p_x with mean $\bar{x} \in \mathbf{R}^n$ and covariance matrix $\Sigma \in \mathbf{S}_+^n$ is given by

$$p_x(x) = \frac{1}{\sqrt{(2\pi)^n \det \Sigma}} \exp\left(-\frac{1}{2}(x - \bar{x})^T \Sigma^{-1}(x - \bar{x})\right).$$

The logarithm of this Gaussian prior is then expressed as

$$\log p_x(x) = -\frac{n}{2}\log(2\pi) - \frac{1}{2}\log\det\Sigma - \frac{1}{2}(x - \bar{x})^T\Sigma^{-1}(x - \bar{x}), \tag{5.31}$$

which is a concave quadratic function of x, so the resulting MAP estimation problem (5.28) is a convex optimization problem. In practice, the mean $\bar{x}$ of the Gaussian

prior is mostly assumed to be zero, and the log-prior term (5.31) can therefore be simplified to

$$\log p_x(x) = -\frac{n}{2}\log(2\pi) - \frac{1}{2}\log\det\Sigma - \frac{1}{2}x^T\Sigma^{-1}x,$$

so the resulting MAP estimation problem (5.28) is given by

$$\text{minimize} \quad \|Ax - y\|_2^2 + x^T\Sigma^{-1}x \tag{5.32}$$

with variable $x \in \mathbf{R}^n$. Note that to obtain this problem formulation we have dropped the constant terms in (5.30) and (5.31) and scaled the two parts in the objective simultaneously by a factor of 2 (and, of course, reversed the sign).

If the covariance matrix Σ is large, *i.e.*, it has large eigenvalues, then the Gaussian prior is relatively flat. As a result, the MAP estimation of x from the problem (5.32) is more close to the maximum likelihood estimation solution with the log-likelihood objective given by (5.30), which is indeed the case since the second term $x^T\Sigma^{-1}x$ in the objective of (5.32) would be small when Σ has large eigenvalues. On the other hand, if Σ is small (in the sense of eigenvalues), then the Gaussian prior is more peaky (and $x^T\Sigma^{-1}x$ would also be large), and hence the MAP estimation of x from the problem (5.32) will be more close to the prior mean zero. (See also exercise 5.4 for more discussion.)

If we further take the assumption that the covariance matrix Σ is a scaled identity matrix $\sigma_x^2 I$, *i.e.*, the components of x are IID Gaussian variables with the same variance σ_x^2, then the problem (5.32) reduces to the Tikhonov regularization least squares problem (5.29) considered in example 5.6. In this case, such a prior essentially encourages the estimation of x to be small in the Euclidean norm, with the strength of the belief about this prior proportional to $1/\sigma_x^2$.

Laplace prior

When the components of the variable $x \in \mathbf{R}^n$ are assumed to be IID Laplace random variables with mean μ and scale parameter $b > 0$, then the log-prior term $\log p_x(x)$ in the MAP estimation problem (5.28) is given by

$$\begin{aligned}
\log p_x(x) &= \log \prod_{i=1}^n \frac{1}{2b}\exp\left(-\frac{|x_i - \mu|}{b}\right) \\
&= \sum_{i=1}^n \left(-\log(2b) - \frac{|x_i - \mu|}{b}\right) \\
&= -n\log(2b) - \frac{1}{b}\|x - \mu\mathbf{1}\|_1.
\end{aligned}$$

Putting this log-prior term together with the log-likelihood term (5.30), the resulting MAP estimation problem (5.28) with this Laplace prior can be expressed as the convex program

$$\text{minimize} \quad \|Ax - y\|_2^2 + (2/b)\|x - \mu\mathbf{1}\|_1$$

with variable $x \in \mathbf{R}^n$. If we further take the assumption that the mean μ of the Laplace distribution is zero, then the problem above reduces to

$$\text{minimize} \quad \|Ax - y\|_2^2 + (2/b)\|x\|_1, \tag{5.33}$$

which has the form of an ℓ_1-norm regularized least squares problem (*i.e.*, the problem (5.16) with primary objective $\|Ax - b\|_2^2$). Therefore, such an IID Laplace prior with zero mean on the variable x is a sparsity-inducing prior.

The distinct impacts of Gaussian and Laplace priors on the MAP estimation problems are readily seen from their density functions. Suppose that we have two IID priors with the same zero mean and variance, one Gaussian and one Laplace, applied to the variable $x \in \mathbf{R}^n$. Since the Laplace distribution has a much sharper peak at zero than the Gaussian distribution (see figure 4.10, page 128), the Laplace prior assigns a much higher probability to the value zero than the Gaussian prior, which hence encourages the estimation of x to be sparse. On the other hand, the heavy tails of the Laplace distribution allows some components of x to take large values when necessary, which is less likely to happen under a Gaussian prior (with the same mean and variance).

Recalling that the variance of a Laplace distribution with scale parameter $b > 0$ is given by $2b^2$, we can see that when b takes a small value, the Laplace prior is more peaky at zero, and hence the resulting MAP estimation of x from (5.33) is more likely to be sparse. This is consistent with the regularization interpretation of (5.33), since the sparisty regularization term $\|x\|_1$ will be associated with a large weight when b takes a small value.

Uniform prior

Suppose that the components of the variable $x \in \mathbf{R}^n$ are IID random variables uniformly distributed over some interval $[-b, b]$ with $b > 0$, then the prior distribution p_x of x can be expressed as

$$p_x(x) = \begin{cases} 1/(2b)^n, & -b\mathbf{1} \preceq x \preceq b\mathbf{1} \\ 0, & \text{otherwise.} \end{cases}$$

By defining $\log 0 = -\infty$, we have

$$\log p_x(x) = \begin{cases} -n \log(2b), & -b\mathbf{1} \preceq x \preceq b\mathbf{1} \\ -\infty, & \text{otherwise,} \end{cases}$$

so the resulting MAP estimation problem (5.28) with the log-likelihood term (5.30) can be expressed as the following convex program

$$\begin{array}{ll} \text{minimize} & \|Ax - y\|_2^2 \\ \text{subject to} & \|x\|_\infty \leq b \end{array}$$

with variable $x \in \mathbf{R}^n$. In other words, applying an IID uniform prior on the interval $[-b, b]$ to the variable x is equivalent to imposing an ℓ_∞-norm constraint on the estimation of x in the MAP estimation problem.

In fact, we can extend this result to a more general case where the variable x is assumed to be uniformly distributed over some set $C \subseteq \mathbf{R}^n$. Since for any $x \in C$ the uniform prior distribution $p_x(x)$ is a constant (equal to the inverse of the volume of C), and $p_x(x) = 0$ for any $x \notin C$, the resulting MAP estimation problem (5.28) with log-likelihood (5.30) resolves to

$$\begin{array}{ll} \text{minimize} & \|Ax - y\|_2^2 \\ \text{subject to} & x \in C \end{array} \tag{5.34}$$

with variable $x \in \mathbf{R}^n$. If the set C can be expressed via convex inequality and equality constraints, then the problem (5.34) is a convex optimization problem.

We can consider an extreme case of (5.34) where the set $C = \mathbf{R}^n$. In this case, the MAP estimation problem reduces to an unconstrained optimization problem with objective $\|Ax - y\|_2^2$, which is essentially the same as the maximum likelihood estimation problem with log-likelihood given by (5.30). We can interpret this observation from the perspective of the prior distribution as follows: Roughly speaking, a uniform prior distribution over the entire space $\mathbf{R}^n$ assigns the same probability to all values of $x \in \mathbf{R}^n$, which means that it does not introduce any preference for any particular value of x, so the resulting MAP estimation problem is equivalent to the maximum likelihood estimation. Here, of course, the function $p_x \colon \mathbf{R}^n \to \mathbf{R}_+$ cannot be expressed as a valid probability distribution such that it integrates to one, but generally takes the form of a (positive) constant function on $\mathbf{R}^n$, and is hence called an *improper prior distribution*.

Prior with restricted support

The *support* of a probability distribution $p \colon \mathbf{R}^n \to \mathbf{R}_+$ is defined as

$$S = \{x \in \mathbf{R}^n \mid p(x) > 0\},$$

i.e., the set of points that have nonzero probability under this distribution. For example, the support of a Gaussian distribution is the entire space $\mathbf{R}^n$, while the support of a uniform distribution over some set C in the problem (5.34) is the set C. When the support of a prior distribution $p_x \colon \mathbf{R}^n \to \mathbf{R}_+$ is some subset of $\mathbf{R}^n$, the resulting MAP estimation problem (5.28) can be interpreted as a constrained optimization problem where the constraint is (implicitly) given by the support of the prior distribution.

This property is interpreted from different perspectives. From a Bayesian perspective, the support of the prior distribution p_x represents the set of values that are considered possible for the variable x according to our prior knowledge, and any value of x outside this support is considered impossible (with zero probability) and hence cannot be a solution to (5.28). From a regularization perspective, the log-prior term $\log p_x(x)$ will introduce an negative infinite penalty to the objective of the MAP estimation problem (5.28) for any value of x outside the support of p_x, which therefore imposes a hard constraint on the estimation of x to be within the support of p_x.

Example 5.7 *Half-normal prior.* The (one-dimensional) *half-normal distribution* is a restriction of the Gaussian distribution with mean zero and variance σ^2 to the

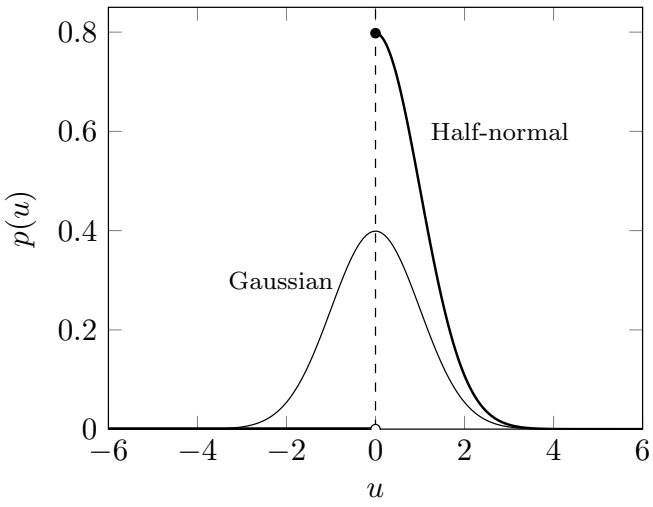

Figure 5.18 Probability density functions of the half-normal distribution given by (5.35) with $\sigma^2 = 1$ (shown thicker) and the standard Gaussian distribution $\mathcal{N}(0, 1)$.

nonnegative orthant $\mathbf{R}_+$, which can be expressed as

$$p(u) = \begin{cases} (2/\pi\sigma^2)^{1/2} \exp(-u^2/(2\sigma^2)), & u \geq 0 \\ 0, & \text{otherwise}, \end{cases} \tag{5.35}$$

where the parameter σ^2 is the variance of the underlying Gaussian distribution. Figure 5.18 shows the probability density function of a half-normal distribution with $\sigma^2 = 1$ in comparison with the standard Gaussian distribution $\mathcal{N}(0, 1)$.

Assuming that the components of the variable $x \in \mathbf{R}^n$ are IID half-normal random variables with the parameter σ^2 in (5.35) equal to 1, then we have

$$\log p_x(x) = \sum_{i=1}^{n} \left(\frac{1}{2} \log(2/\pi) - \frac{x_i^2}{2} \right) = \frac{n}{2} \log(2/\pi) - \frac{1}{2} \|x\|_2^2$$

for any $x \succeq 0$, and $\log p_x(x) = -\infty$ otherwise. Therefore, the resulting MAP estimation problem (5.28) with log-likelihood (5.30) and this half-normal prior can be expressed as

$$\begin{array}{ll} \text{minimize} & \|Ax - y\|_2^2 + \|x\|_2^2 \\ \text{subject to} & x \succeq 0 \end{array}$$

with variable $x \in \mathbf{R}^n$, which is the Tikhonov regularized least squares problem with a nonnegativity constraint.

Numerical example

We present a simple numerical example to summarize the discussions above. Consider an MAP estimation problem (5.28) with the log-likelihood term given by

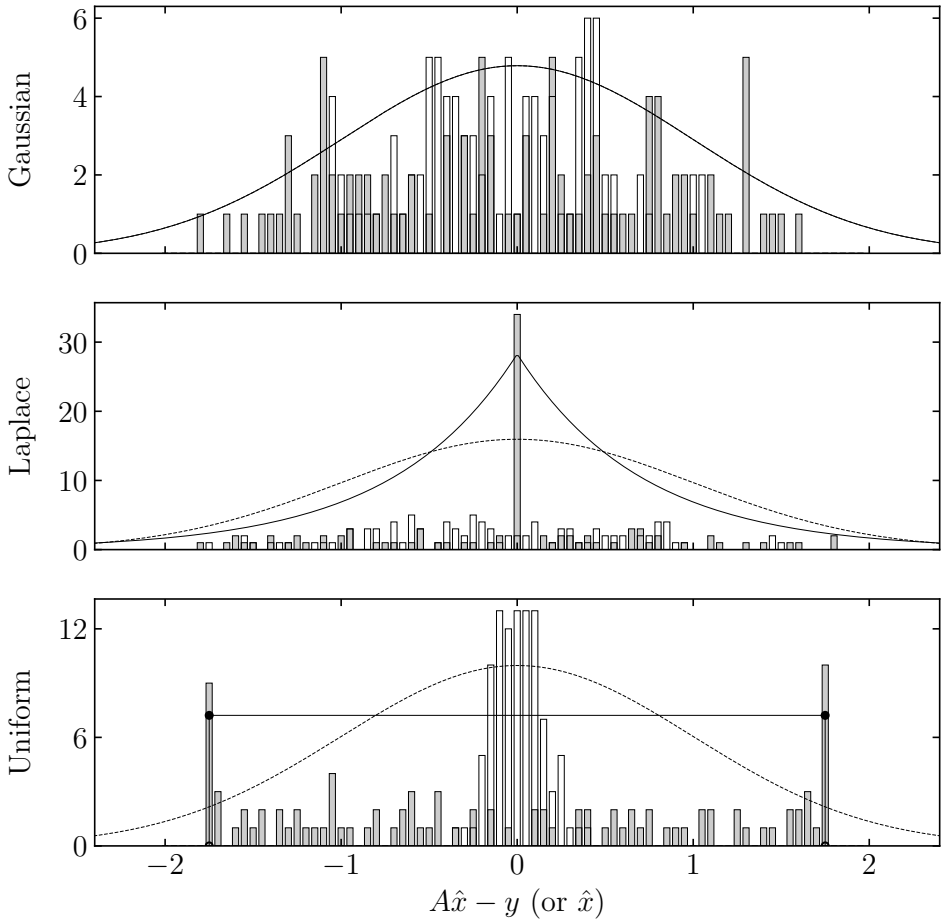

Figure 5.19 Histograms of the component amplitudes of the estimation $\hat{x}$ (shaded bars) and the corresponding residual $A\hat{x} - y$ (empty bars) obtained from solving the MAP estimation problem (5.28) with log-likelihood (5.30) and different choices of the prior distribution. The solid curves in each plot show the (scaled) probability density functions of the corresponding (componentwise) prior distributions, and the standard Gaussian distribution $\mathcal{N}(0, 1)$ (which is assumed for the linear measurement noise) is shown as the dashed curve for reference.

(5.30), and the components of the variable $x \in \mathbf{R}^n$ are assumed to be IID according to one of the following three priors:

- Standard Gaussian $\mathcal{N}(0, 1)$.
- Laplace with mean zero and scale parameter $b = 1/\sqrt{2}$.
- Uniform over the interval $[-\sqrt{3}, \sqrt{3}]$.

Note that all these three priors have the same mean zero and variance 1. Figure 5.19 shows the histograms for the component amplitudes of the MAP estimation $\hat{x}$ and the corresponding residual $A\hat{x} - y$ under these setups, where the problem data $A \in \mathbf{R}^{100\times 100}$ and $y \in \mathbf{R}^{100}$ are generated randomly.

We have the following observations from figure 5.19.

- When the prior is the standard Gaussian distribution $\mathcal{N}(0, 1)$, the estimation $\hat{x}$ has a similar amplitude distribution as the residual $A\hat{x} - y$, since the prior distribution for x has the same variance as the prior distribution for the residual.
- Under the Laplace prior, many entries of $\hat{x}$ are exactly zero, since the Laplace distribution has a much sharper peak at zero than the Gaussian distribution which therefore encourages sparsity in the estimation. On the other hand, the optimal residual amplitude distribution is more widely spread out than the previous case.
- When the prior is uniform on $[-\sqrt{3}, \sqrt{3}]$, all entries of $\hat{x}$ are bounded by the interval $[-\sqrt{3}, \sqrt{3}]$, and most of the amplitudes are close to the boundary values $\pm\sqrt{3}$, since according to this prior distribution, the estimation $\hat{x}$ is not allowed to take values outside its support $[-\sqrt{3}, \sqrt{3}]$.

5.5 Matrix regularizers

The ideas of regularization can be extended to matrix variables. Specifically, let $\phi\colon \mathbf{R}^{m\times n} \to \mathbf{R}$ be a regularization function defined on the space of $m \times n$ matrices, then we can consider the regularized problem

$$\text{minimize} \quad f_0(X) + \gamma\phi(X)$$

with variable $X \in \mathbf{R}^{m\times n}$, where $f_0\colon \mathbf{R}^{m\times n} \to \mathbf{R}$ is the primary objective, $\gamma > 0$ is the regularization weight, and $\phi(X)$ is the regularization term that incorporates our prior knowledge about the desired characteristics on X.

5.5.1 Componentwise regularizers

Some strategies of vector regularization can be readily generalized to matrix variables by applying the same penalty to each entry of the matrix variable X. For

example, the generalization of vector Tikhonov regularization on a matrix variable $X \in \mathbf{R}^{m\times n}$ is given by

$$\phi(X) = \|X\|_F^2 = \sum_{i=1}^{m}\sum_{j=1}^{n} X_{ij}^2, \tag{5.36}$$

where $\|\cdot\|_F$ is the Frobenius norm of a matrix. Similarly, the *sum-absolute-value norm* regularization of X is given by

$$\phi(X) = \|X\|_{\text{sav}} = \sum_{i=1}^{m}\sum_{j=1}^{n} |X_{ij}|, \tag{5.37}$$

which can be regarded as a matrix version of the ℓ_1-norm regularization for vector variables.

From the examples above, it is easily seen that applying a componentwise regularization to a matrix variable is equivalent to applying the corresponding vector regularization function to the vector obtained from flattening the matrix by, *e.g.*, a column-priority ordering. Hence, the expected solution characteristics of matrix regularization problems with componentwise regularizers are essentially the same as those of the corresponding vector regularization problems. Specifically, the squared Frobenius norm regularization (5.36) encourages the solution matrix to have small entries, while the sum-absolute-value norm regularization (5.37) encourages the solution matrix to be sparse in terms of its entries.

5.5.2 Columnwise sparsity

Another useful matrix regularization strategy is to encourage the solution matrix to have *columnwise sparsity*, *i.e.*, many columns of the matrix are zero vectors. Specifically, let

$$X = \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix} \in \mathbf{R}^{m\times n},$$

be the matrix variable, where $x_i \in \mathbf{R}^m$ is the ith column of X, then the regularization function

$$\phi(X) = \sum_{i=1}^{n} \|x_i\|_2 \tag{5.38}$$

encourages X to have columnwise sparsity. Importantly, we should note that there is no square in the regularization term $\|x_i\|_2$ for each column x_i, since otherwise the resulting regularization function ϕ would be the same as the squared Frobenius norm regularization (5.36), which simply encourages the solution matrix to have small entries instead of columnwise sparsity.

The sparsity-inducing property of the regularization (5.38) can be interpreted as follows. First we notice that the ℓ_1-norm $\|u\|_1$ for some nonnegative vector $u \in \mathbf{R}_+^n$ is simply the sum of the entries of u. Since we have $\|x_i\|_2 \geq 0$ for all $i = 1, \ldots, n$, the regularization function $\phi(X)$ in (5.38) can be regarded as the ℓ_1-norm of the vector of column norms of X, which therefore encourages many of these column norms to be zero, and hence encourages many columns of X to be zero vectors.

These ideas are readily extended to inducing *rowwise sparsity* or *block sparsity* of some matrix variable $X \in \mathbf{R}^{m\times n}$ by applying the same ℓ_1-norm regularization to the vector of row or block matrix norms of X; see exercise 5.5.

Example 5.8 *Group lasso regression.* Suppose we are given the data $A_i \in \mathbf{R}^{m\times n}$ for $i = 1, \ldots, k$ and $b \in \mathbf{R}^k$, and we want to (approximately) solve the system of linear equations

$$\mathbf{tr}(A_i^T X) = b_i, \quad i = 1, \ldots, k, \tag{5.39}$$

with variable $X \in \mathbf{R}^{m\times n}$. For this purpose, we may consider the primary objective function $f_0 \colon \mathbf{R}^{m\times n} \to \mathbf{R}$ given by

$$f_0(X) = \sum_{i=1}^{k} \left(\mathbf{tr}(A_i^T X) - b_i\right)^2.$$

Minimizing the function f_0 alone can be considered as a matrix version of the least squares linear approximation problem (4.6) on page 108. The difference here is that the features of each data point is given by a matrix $A_i \in \mathbf{R}^{m\times n}$. Let

$$A_i = \begin{bmatrix} a_{i1} & \cdots & a_{in} \end{bmatrix} \quad \text{and} \quad X = \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix},$$

where $a_{ij}, x_j \in \mathbf{R}^m$ forms the jth column of A_i and X, respectively. Then the function f_0 can be rewritten as

$$f_0(X) = \sum_{i=1}^{k} \left(\sum_{j=1}^{n} a_{ij}^T x_j - b_i\right)^2.$$

Hence, we can interpret each column a_{ij} of the data matrix A_i as the jth *feature group* of the ith data point, and the vector x_j is then the corresponding coefficient (vector).

Now suppose we have the prior knowledge that only a few feature groups a_{ij} are relevant to the system of linear equations (5.39), which means that only a few columns of the coefficient matrix X are expected to be nonzero vectors. Putting this prior knowledge together with the primary objective f_0 described above, we have the regularized approximation problem

$$\text{minimize} \quad \sum_{i=1}^{k} \left(\sum_{j=1}^{n} a_{ij}^T x_j - b_i\right)^2 + \gamma \sum_{i=1}^{n} \|x_i\|_2 \tag{5.40}$$

with variable $X \in \mathbf{R}^{m\times n}$, where $\gamma > 0$ is the regularization weight. This problem is a convex optimization problem and can be considered as a matrix version of the lasso regression problem in the form (5.16). Here, the required sparsity pattern is on the feature groups, rather than individual (scalar) features. As a result, the problem (5.40) is sometimes called the *group lasso regression*.

We can compare the solution characteristics induced by the columnwise sparsity regularization (5.38) with those induced by the sum-absolute-value norm regularization (5.37), which encourages componentwise sparsity in the matrix variable X. As a specific example, we consider the problem (5.40) from the example above and the problem

$$\text{minimize} \quad \sum_{i=1}^{k} \left(\sum_{j=1}^{n} a_{ij}^T x_j - b_i\right)^2 + \gamma \|X\|_{\text{sav}}, \tag{5.41}$$

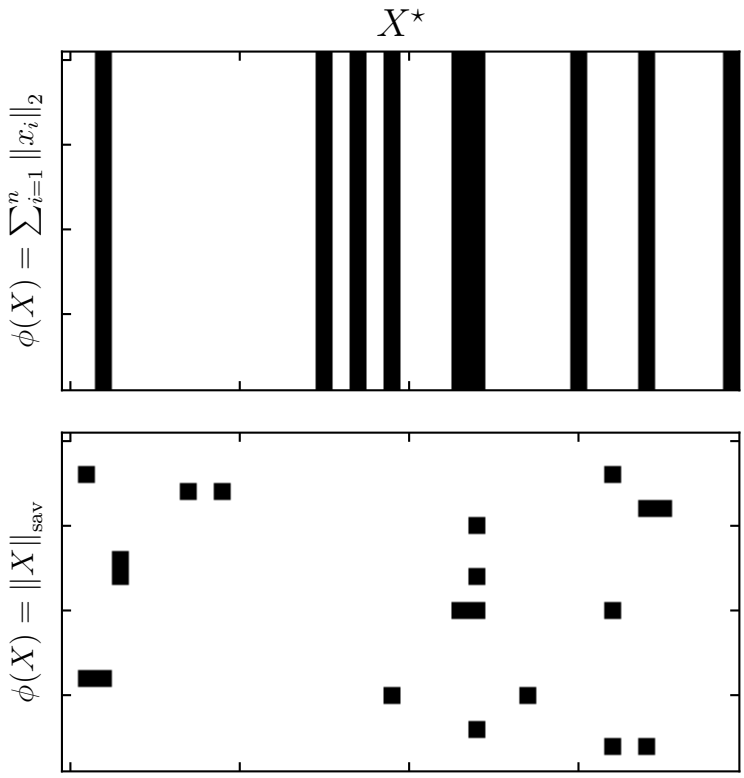

Figure 5.20 Sparsity patterns of solution $X^\star$ of the problems (5.40) (shown top) and (5.41) (shown bottom) on an example dataset. In the shown results, the obtained solutions from both problems lead to the same primary objective value. The nonzero entries of the solution matrix are shown in black and the zero entries are shown in white.

which has the same primary objective as (5.40) but a different regularization term given by (5.37). Figure 5.20 shows the solution sparsity patterns under these two regularization strategies, where the randomly generated data $A_i \in \mathbf{R}^{m\times n}$ for $i = 1,\ldots,k$ and $b \in \mathbf{R}^k$ have dimension $m = 20$, $n = 40$, and $k = 20$. The nonzero entries of the solution matrix are shown in black and the zero entries are shown in white. In the top figure, the solution obtained from the problem (5.40) with regularization (5.38) exihibits a clear columnwise sparsity pattern. In contrast, the solution under the sum-absolute-value norm regularization (5.37) shown in the bottom is more scattered and less structured.

5.5.3 Rank regularization

A generalization of the cardinality regularization for vector variables is the *rank regularization* for matrix variables, which encourages the solution matrix to have low rank. Specifically, a rank regularized problem with matrix variable $X \in \mathbf{R}^{m\times n}$ can be expressed as

$$\text{minimize} \quad f_0(X) + \gamma\,\mathbf{rank}\,X, \tag{5.42}$$

where $f_0\colon \mathbf{R}^{m\times n} \to \mathbf{R}$ is the primary objective, $\gamma > 0$ is the regularization weight, and $\mathbf{rank}\,X$ is the rank of the matrix X.

Since the rank regularization term $\mathbf{rank}\,X$ is a nonconvex function of X, the problem (5.42) is usually very hard to solve. However, a good heuristic for getting an approximate solution to the problem (5.42) is to replace the rank regularization with the *nuclear norm* regularization given by

$$\phi(X) = \|X\|_* = \sum_{i=1}^{\min\{m,n\}} \sigma_i(X),$$

where $\sigma_i(X)$ is the ith largest singular value of X (see also §A.3.2, page 354). Similar to the ℓ_1-norm for the cardinality function, the nuclear norm is the convex envelope of the rank function (under some technical conditions, see example 2.11), and hence the problem

$$\text{minimize} \quad f_0(X) + \gamma\|X\|_* \tag{5.43}$$

with variable $X \in \mathbf{R}^{m\times n}$ is a convex approximation of the rank regularized problem (5.42). In practice, solving the problem (5.43) usually gives a good (and sometimes quite good) approximation to the solution of (5.42).

Remark 5.2 *Interpretation via singular value sparsity.* The relationship between the rank function and the nuclear norm of some matrix $X \in \mathbf{R}^{m\times n}$ can also be intuitively understood from the perspective of the sparsity of the singular values of X. Specifically, $\mathbf{rank}\,X$ is equal to the *number* of nonzero singular values of X, while $\|X\|_*$ is equal to the *sum* of the singular values of X. Hence, the rank regularization encourages the solution matrix to have a small number of nonzero singular values, while the nuclear norm regularization encourages the solution matrix to have a small sum of singular values, which indirectly promotes low rank solutions. This is similar to the relationship between the cardinality function and the ℓ_1-norm for some vector

$x \in \mathbf{R}^n$, where $\mathbf{card}\, x$ counts the number of nonzero entries of x, and $\|x\|_1$ sums up the absolute values of all its components.

Example 5.9 *Low rank matrix completion.* One example of the rank regularization problem (5.42) is the *low rank matrix completion* problem, where we want to find a low rank matrix that is consistent with a partially observed data matrix. Specifically, let $A \in \mathbf{R}^{m\times n}$ be some matrix, and we are given a corrupted observation A^{obs} of A. Moreover, some entries of the true matrix A are missing in the observation A^{obs}, where $\mathcal{I} \subseteq \{1,\ldots,m\} \times \{1,\ldots,n\}$ is the index set denoting which entries are observed. The low rank matrix completion problem can be expressed as

$$\text{minimize} \quad \sum_{(i,j)\in\mathcal{I}} \left(X_{ij} - A^{\mathrm{obs}}_{ij}\right)^2 + \gamma\, \mathbf{rank}\, X, \tag{5.44}$$

where the variable $X \in \mathbf{R}^{m\times n}$ is the full matrix we want to find. The primary objective of this problem is to find a matrix X with complete entries that is consistent with the noisy observation as much as possible, and the regularization term encourages the solution matrix to have low rank. A convex approximation of the problem (5.44) in the nuclear norm regularized form (5.43) is given by

$$\text{minimize} \quad \sum_{(i,j)\in\mathcal{I}} \left(X_{ij} - A^{\mathrm{obs}}_{ij}\right)^2 + \gamma\|X\|_*. \tag{5.45}$$

Figure 5.21 shows an example of the low rank matrix completion problem, where the true matrix $A \in \mathbf{R}^{100\times 100}$ with $\mathbf{rank}\, A = 2$ is shown on the left, and the observed matrix A^{obs} (middle) is corrupted by a Gaussian noise with roughly 60% entries missing. A complete recovery $X^\star$ of the true matrix A obtained from the nuclear norm heuristic (5.45) is shown on the right. It can be seen that even with only 40% of the entries observed, the estimation $X^\star$ recovers the gradient pattern of the true matrix A quite well. The first 20 singular values of the matrices A, A^{obs}, and $X^\star$ are shown in figure 5.22 as dashed, dashdotted, and solid lines, respectively. The true matrix A has two nonzero singular values (since $\mathbf{rank}\, A = 2$), while the observed matrix A^{obs} is almost full rank due to the Gaussian corruption. Notably, the first two singular values of the recovery $X^\star$ from (5.45) are close to those of the true matrix A, while the rest of its singular values are almost zero.

See also §8.3.4 for more discussions on matrix completion problems.

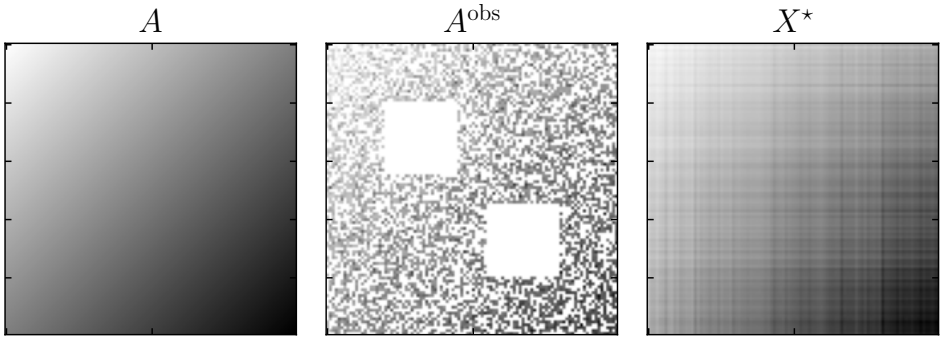

Figure 5.21 *Low rank matrix completion.* The true matrix A (left), the observed matrix A^{obs} (middle), and a complete recovery $X^\star$ obtained from the nuclear norm heuristic (5.45) (right).

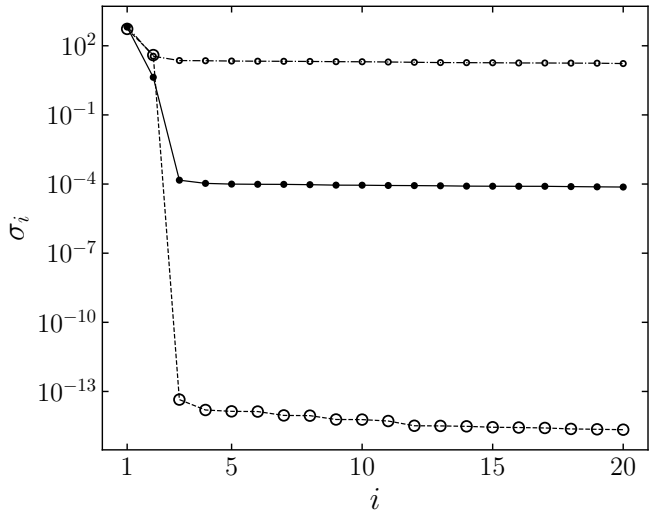

Figure 5.22 The first 20 singular values of the matrices A, A^{obs}, and $X^\star$ in figure 5.21, shown as dashed, dashdotted, and solid lines, respectively.

Bibliographical notes

In the most general case, a vector optimization problem can be defined as minimizing a vector-valued objective function $f_0 \colon \mathbf{R}^n \to \mathbf{R}^k$ with respect to a generalized inequality induced by some *proper cone* (see [BV04, §2.4.1]). The multiobjective optimization problem (5.1) considered in this chapter is a special case where the generalized inequality is defined by the nonnegative orthant $\mathbf{R}^k_+$. Some properties, *e.g.*, *minimum* and *minimal elements* with respect to a generalized inequality, related to the most general form of vector optimization problems can be found in [BV04, §2.6.3 and §4.7].

Vector optimization originated from the field of economics. In particlar, the term *Pareto optimality* for multiobjective problems was first introduced by the Italian economist Vilfredo Pareto in 1906 [Par14]. See also Debreu [Deb59] and Mas-Colell *et al.* [MWG95].

Regularized approximations are covered in many books. See, *e.g.*, Tikhonov and Arsenin [TA77], Engl *et al.* [EHN96], Hansen [Han98, Han10], and Golub and Van Loan [GV13] for Tikhonov regularization and ridge regression, and Hastie *et al.* [HTF09] for lasso and regressor selection.

The concept of total variation in mathematics was initially introduced by Jordan in the analysis of Fourier series [Jor81]. It was later used in the context of image processing and denoising by Rudin *et al.* [ROF92] and Chambolle [Cha04].

For more discussions on maximum a posteriori estimation, see, *e.g.*, Murphy [Mur12], as well as the other references listed on page 140.

Matrix regularization methods based on grouped penalties, *e.g.*, the group lasso, was introduced in Yuan and Lin [YL06] and Bach *et al.* [BJMO12]. Rank regularization and the nuclear norm heuristic are covered in Fazel's PhD thesis [Faz02] and Fazel *et al.* [FHB03]. Low-rank matrix completion with theoretical recovery guarantees is treated in Candès and Tao [CT10], Keshavan *et al.* [KMO10], and Candès and Recht [CR09, CR12]. For practical large-scale solvers of matrix completion problems via the *singular value thresholding*, see Cai *et al.* [CCS10].

Exercises

Multiobjective optimization

5.1 Let $\mathcal{O}$ be the set of achievable objective values of some multiobjective optimization problem in the form (5.1). Consider the set

$$\mathcal{A} = \mathcal{O} + \mathbf{R}^k_+ = \{t \in \mathbf{R}^k \mid f_0(x) \preceq t \text{ for some feasible } x\},$$

which consists of all values that are worse than or equal to some achievable objective value. Show that the set $\mathcal{A}$ is a convex set if the original multiobjective optimization problem is convex.

Regularized approximation

5.2 [BV04, page 307] *Multiple regularized approximation.* Consider an optimal input design problem for a dymical system with scalar input sequence $u = (u(0), u(1), \ldots, u(n))$, and scalar output sequence $y = (y(0), y(1), \ldots, y(n))$. The input and output sequences are related by the linear convolution

$$y(t) = \sum_{\tau=0}^{t} h(\tau)u(t-\tau), \quad t = 0, \ldots, n,$$

where $h(0), h(1), \ldots, h(n)$ is the *convolution kernel* or *impulse response* of the system. The goal of the optimal input design problem is to find an input sequence u that achieves the following objectives.

- *Output tracking.* The primary goal is to make the output sequence y close to some desired output sequence y_{des}.
- *Small input.* The secondary goal is to keep the input sequence u small in some appropriate norm.
- *Smooth input.* The third goal is to make the input sequence u smooth, *i.e.*, it should not vary rapidly between consecutive time steps.

(a) Let J_{track}, J_{mag}, and J_{var} be the cost functions corresponding to the three objectives above, respectively. Propose some specific forms of these cost functions that are suitable for the optimal input design problem described above, and formulate the optimization problem for this optimal input design task.

(b) Consider a specific numerical example of the optimal input design problem that you formulated in (a), with $n = 200$. The impulse response of the system is given by

$$h(t) = \frac{1}{9}(0.9)^t(1 - 0.4\cos(2t)), \quad t = 0, \ldots, n,$$

and the desired output sequence is given by the following square wave:

$$y_{\text{des}}(t) = \begin{cases} 0, & t \in [0, 50) \cup [150, 200) \\ 1, & t \in [50, 100) \\ -1, & t \in [100, 150) \end{cases}$$

for $t = 0, \ldots, n$. Solve the optimal input design problem for this example, and plot the obtained optimal input sequence u and the corresponding output sequence y for different trade-offs between the three objectives.

Smoothing

5.3 *Curvature smoothing.* Suppose that the vector $x \in \mathbf{R}^n$ represents a one-dimensional time series, and we want to find a smooth estimation $\hat{x}$ of x based on some noisy observation x_{obs} via the smoothing problem (5.19). Instead of using the first-order difference $\hat{x}_{i+1} - \hat{x}_i$ to measure the smoothness of the estimation $\hat{x}$, we consider the second-order difference

$$(\hat{x}_{i+1} - \hat{x}_i) - (\hat{x}_i - \hat{x}_{i-1}) = \hat{x}_{i+1} - 2\hat{x}_i + \hat{x}_{i-1}$$

which is a discrete approximation of the *curvature* of the signal. Define the tradiagonal matrix $\Delta \in \mathbf{R}^{(n-2)\times n}$ as

$$\Delta = \begin{bmatrix} 1 & -2 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -2 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & -2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 & -2 & 1 \end{bmatrix},$$

then the (quadratic) curvature regularization of some estimation $\hat{x}$ is given by

$$\phi(\hat{x}) = \|\Delta\hat{x}\|_2^2 = \sum_{i=2}^{n-1} (\hat{x}_{i+1} - 2\hat{x}_i + \hat{x}_{i-1})^2.$$

(a) What are the extreme Pareto optimal values and points of the smoothing problem (5.19) with the curvature regularization $\|\Delta\hat{x}\|_2^2$?

(b) Consider an observation of some time series $x_{\text{obs}} = x + v \in \mathbf{R}^{100}$, given by the true signal $x = 0.5t$ for $t = 1, \ldots, 100$ plus the IID Gaussian random noise $v \in \mathbf{R}^{100}$ where $v_i \sim \mathcal{N}(0, 1)$ for all $i = 1, \ldots, 100$. Plot the estimation $\hat{x}$ obtained from solving the scalarized smoothing problem (5.20) with the curvature regularization $\|\Delta\hat{x}\|_2^2$ for different values of $\gamma > 0$. How does the estimation $\hat{x}$ change as we increase the value of γ? How is the estimation $\hat{x}$ different from the one obtained from solving the same problem with the first-order difference regularization $\phi(\hat{x}) = \sum_{i=1}^{n-1} (\hat{x}_{i+1} - \hat{x}_i)^2$?

Maximum a posteriori estimation

5.4 *MAP estimation with Gaussian prior.* Consider the MAP estimation problem (5.32), where the prior distribution p_x for the variable $x \in \mathbf{R}^n$ is some Gaussian distribution with mean zero.

(a) What is the optimal point of this problem?

(b) Suppose the covariance matrix is diagonal, *i.e.*, $\Sigma = \mathbf{diag}(\sigma_1^2, \ldots, \sigma_n^2)$, where σ_i^2 is the variance of the ith component of the variable x. How does the value of σ_i^2 influence the estimation of x in the problem (5.32)?

(c) Let $\Sigma_1, \Sigma_2 \in \mathbf{S}_+^2$ given by

$$\Sigma_1 = \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad \Sigma_2 = \begin{bmatrix} 2.5 & 1.5 \\ 1.5 & 2.5 \end{bmatrix}$$

be the covariance matrices of two Gaussian priors for the variable $x \in \mathbf{R}^2$.

i. What are the eigenvalues of Σ_1 and Σ_2?

ii. Plot the 90% probability level curve in $\mathbf{R}^2$ for these two Gaussian priors.

iii. Let $\hat{x}_1$ and $\hat{x}_2$ be two estimations of x from the problem (5.32), where the covariance matrices of the Gaussian priors are given by Σ_1 and Σ_2, respectively. Use the plot from ii. to discuss how $\hat{x}_1$ and $\hat{x}_2$ might look like. What are the similarities and differences between these two estimations?

Matrix regularizers

5.5 *Block sparsity regularization.* Let $X \in \mathbf{R}^{m\times n}$ be a matrix variable, and suppose it is partitioned into p blocks as

$$X = \left[\begin{array}{ccc} X_1 & \cdots & X_p \end{array} \right],$$

where each block X_i is a submatrix of X with some fixed size. Define a suitable regularization function ϕ that encourages block sparsity of X, *i.e.*, many of the blocks X_i are zero matrices. Explain why it works.

Chapter 6

Constraints

6.1 Optimization with constraints

Consider an optimization problem of the form

$$\text{minimize} \quad f_0(x) \tag{6.1}$$

where $x \in \mathbf{R}^n$ is the variable and $f_0 \colon \mathbf{R}^n \to \mathbf{R}$ is the objective function. In many applications, we might want to add *constraints* to the problem (6.1), which leads to a *constrained optimization problem* of the form

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \ldots, m \\ & h_i(x) = 0, \quad i = 1, \ldots, p, \end{array} \tag{6.2}$$

where $x \in \mathbf{R}^n$ is the variable, $f_i \colon \mathbf{R}^n \to \mathbf{R}$ for $i = 1, \ldots, m$ are the inequality constraint functions, and $h_i \colon \mathbf{R}^n \to \mathbf{R}$ for $i = 1, \ldots, p$ are the equality constraint functions. If the functions f_i, $i = 0, \ldots, m$, are convex and the functions h_i, $i = 1, \ldots, p$, are affine, then the problem (6.2) is a convex optimization problem.

We also occasionally consider the abstract form of a constrained optimization problem, expressed as

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & x \in \mathcal{X}, \end{array} \tag{6.3}$$

where $\mathcal{X} \subseteq \mathbf{R}^n$ is the *constraint set*. By taking

$$\mathcal{X} = \left\{ x \in \mathbf{R}^n \;\middle|\; \begin{array}{l} f_i(x) \leq 0, \; i = 1, \ldots, m \\ h_i(x) = 0, \; i = 1, \ldots, p \end{array} \right\},$$

we can express the problem (6.2) in the abstract form (6.3).

6.1.1 Interpretations

Generally speaking, a constraint is a condition that the variable $x \in \mathbf{R}^n$ must satisfy. Then the goal of a constrained optimization problem in the form (6.2) is to

find a point that satisfies *all* the constraints (*i.e.*, a *feasible point*), while minimizing the objective function. In other words, it is completely unacceptable for a solution point to violate any of the constraints, even if it may leads to a smaller objective value than any feasible point.

Constraints can be introduced for various reasons, *e.g.*,

- A constraint may be a physical property that must be satisfied. For example, the concentration of a chemical substance must be nonnegative, and the probability of an event must be between 0 and 1.
- A constraint may be a safety requirement that must be guaranteed, such as the maximum load of a bridge or the maximum speed of a vehicle.
- A constraint may be a budget limit, *e.g.*, the maximum amount of money that can be spent on a project or the maximum amount of time that can be allocated to a task.

In the most general case, a constraint can be interpreted as some kind of *prior knowledge* about the problem variable x, that we have a very strong belief in its validity.

Constraints and regularization

Constraints and regularization functions are two options for incorporating prior knowledge into an optimization problem, where their difference is that, roughly speaking, a constraint is a 'hard' prior that must be satisfied, while a regularization is a 'soft' prior that can be violated at some cost.

Example 6.1 As a simple example, consider the Tikhonov regularization least squares problem

$$\text{minimize} \quad \|Ax - b\|_2^2 + \lambda\|x\|_2^2, \tag{6.4}$$

where $x \in \mathbf{R}^n$ is the variable, $A \in \mathbf{R}^{m\times n}$ and $b \in \mathbf{R}^m$ are given data. The value of the hyperparameter $\lambda > 0$ represents the strength of our expectation that x should be close to the zero vector in the sense of the ℓ_2-norm. If we are not quite sure about how small x should be, we can solve the problem (6.4) with a small value of λ, and then increase λ until we get a solution that satisfies our expectation. On the other hand, if we are very certain that the Euclidean distance of x to the zero should be at most t (with $t > 0$), we can directly solve the following constrained optimization problem

$$\begin{array}{ll} \text{minimize} & \|Ax - b\|_2^2 \\ \text{subject to} & \|x\|_2^2 \leq t \end{array} \tag{6.5}$$

with variable $x \in \mathbf{R}^n$. Similar to the problem (6.4), the problem (6.5) is a convex optimization problem, and a smaller value of t forces the solution to be closer to the zero vector.

These ideas are readily adapted to the other types of regularization functions presented in chapter 5.

We will see later in §6.7 that the relationship between constraints and regularization can be very useful when dealing with infeasible problems, *i.e.*, there is no point that satisfies all the constraints.

6.1.2 Examples of constraints

As some basic examples, this section presents several types of the most fundamental constraints that frequently appear in practice. We will see more examples of constraints in later sections of this chapter (and also in part III of the book), when we discuss more specific applications.

Nonnegativity constraints

An optimization problem with a *nonnegativity constraint* of the variable $x \in \mathbf{R}^n$ has the form

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & x \succeq 0. \end{array} \tag{6.6}$$

The nonnegativity constraint $x \succeq 0$ means that every component of the variable x must be nonnegative, *i.e.*, $x_i \geq 0$ for all $i = 1, \ldots, n$. This type of constraint has many applications, *e.g.*, when the variable x represent some physical quantity that cannot be negative, such as concentration, population, intensity, power, etc. The nonnegativity constraint in (6.6) is a convex (specifically, linear) inequality constraint, so if the objective function f_0 is convex, then the problem (6.6) is a convex optimization problem.

Example 6.2 *Nonnegative least squares.* Consider the following optimization problem

$$\begin{array}{ll} \text{minimize} & \|Ax - b\|_2^2 \\ \text{subject to} & x \succeq 0, \end{array} \tag{6.7}$$

where $x \in \mathbf{R}^n$ is the variable, and $A \in \mathbf{R}^{m \times n}$ and $b \in \mathbf{R}^m$ are given data. This problem is called the *nonnegative least squares* problem, and is convex. We can interpret the problem (6.7) as finding the best approximation of the vector b (under the Euclidean norm) from all nonnegative linear combinations (*i.e.*, *conic combinations*) of the column vectors of A. Hence, the problem (6.7) consists in finding a (Euclidean) projection of the vector b onto the convex cone generated by the column vectors of the matrix A.

The idea of nonnegativity constraint for vectors can be extended to *matrix* variables, where we can, *e.g.*, require that every entry of the matrix variable is nonnegative, *i.e.*,

$$\begin{array}{ll} \text{minimize} & f_0(X) \\ \text{subject to} & X_{ij} \geq 0, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n, \end{array}$$

where $X \in \mathbf{R}^{m \times n}$ is the matrix variable.

Example 6.3 *Nonnegative matrix factorization.* Let $A \in \mathbf{R}^{m \times n}$ be a given data matrix, and we want to find two nonnegative matrices $X \in \mathbf{R}_+^{m \times k}$ and $Y \in \mathbf{R}_+^{k \times n}$ such that their product XY is as close to A as possible. This problem can be formulated as

$$\begin{array}{ll} \text{minimize} & \|XY - A\|_F^2 \\ \text{subject to} & X_{ij} \geq 0, \quad i = 1, \ldots, m, \quad j = 1, \ldots, k \\ & Y_{ij} \geq 0, \quad i = 1, \ldots, k, \quad j = 1, \ldots, n \end{array} \tag{6.8}$$

with variables X and Y, where $\|\cdot\|_F$ is the Frobenius norm, *i.e.*, the square root of the sum of squares of all entries of a matrix. The problem (6.8) is sometimes called the *nonnegative matrix factorization* problem.

It is obviously seen that the problem (6.8) is not convex, since the objective function is not jointly convex in X and Y. However, we may notice that the problem (6.8) is convex in X when Y is fixed, and is convex in Y when X is fixed, so it is a *biconvex* optimization problem and can be approximately solved via alternating minimization. (See also example 3.4.)

On the other hand, we can also require that the matrix variable is *positive semidefinite*, *i.e.*,

$$\begin{array}{ll} \text{minimize} & f_0(X) \\ \text{subject to} & X \succeq 0, \end{array}$$

where $X \in \mathbf{S}^n$ is a symmetric matrix variable. Here the generalized inequality constraint $X \succeq 0$ means that

$$X \in \{X \in \mathbf{S}^n \mid v^T X v \geq 0 \text{ for all } v \in \mathbf{R}^n\}.$$

We have seen an application of such constraint in the problem of Gaussian covariance estimation (4.27) on page 133.

Box constraints

An optimization problem with a *box constraint* on the variable $x \in \mathbf{R}^n$ has the form

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & l \preceq x \preceq u, \end{array} \tag{6.9}$$

where $l, u \in \mathbf{R}^n$ are given vectors that specify the lower and upper bounds of the variable x, respectively. This type of constraint requires that every component of the variable $x \in \mathbf{R}^n$ must be between the corresponding components of l and u, *i.e.*, $l_i \leq x_i \leq u_i$ for all $i = 1, \ldots, n$.

If the function f_0 corresponds to a linear norm approximation objective, *i.e.*, $f_0(x) = \|Ax - b\|$ for some given data $A \in \mathbf{R}^{m \times n}$ and $b \in \mathbf{R}^m$, then the resulting box-constrained approximation problem

$$\begin{array}{ll} \text{minimize} & \|Ax - b\| \\ \text{subject to} & l \preceq x \preceq u \end{array}$$

consists in projecting the vector b (with respect to the norm $\|\cdot\|$) onto the image of the box $\{x \in \mathbf{R}^n \mid l \preceq x \preceq u\}$, under the linear transformation defined by the matrix A.

The box constraint in (6.9) includes many useful constraints as special cases. For example, the nonnegativity constraint $x \succeq 0$ corresponds to taking $l = 0$ and $u = +\infty$; if we take $l = -\infty$ and $u = +\infty$, then the problem (6.9) reduces to the unconstrained case (6.1). Moreover, by taking $l = \alpha\mathbf{1}$ and $u = \beta\mathbf{1}$ with $\alpha, \beta \in \mathbf{R}$ ($\alpha \leq \beta$), we can require that every component of the variable x lies in the interval

$[\alpha, \beta]$. Last but not least, if we choose $l, u \in \mathbf{R}^n$ such that $-l = u = t\mathbf{1}$ with $t \geq 0$, then the problem (6.9) is equivalent to the ℓ_∞-norm constrained problem

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & \|x\|_\infty \leq t \end{array} \tag{6.10}$$

with variable $x \in \mathbf{R}^n$. These ideas also generalize to matrix variables.

Probability constraints

Adding a simple equality constraint to the nonnegativity constrained problem (6.6), we obtain an optimization problem with a *probability constraint* of the form

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & x \succeq 0, \quad \mathbf{1}^T x = 1, \end{array} \tag{6.11}$$

where $x \in \mathbf{R}^n$ is the variable. The inequality and inequality constraints in (6.11) together require that the variable x must be a nonnegative vector whose entries sum up to 1, *i.e.*, the variable $x \in \mathbf{R}^n$ defines a probability distribution over a set of n elements. Geometrically, this type of constraint restricts the variable x to lie in the *probability simplex* in $\mathbf{R}^n$ (see example 2.3, page 29). If the objective function f_0 is convex, then the problem (6.11) is a convex optimization problem.

We have seen in §4.3 that probability constraints are often associated with distribution estimation problems. They can also appear in approximation problems. For example, consider the linear norm approximation problem

$$\begin{array}{ll} \text{minimize} & \|Ax - b\| \\ \text{subject to} & x \succeq 0, \quad \mathbf{1}^T x = 1, \end{array}$$

where $x \in \mathbf{R}^n$ is the variable, $A \in \mathbf{R}^{m \times n}$ and $b \in \mathbf{R}^m$ are given data, and $\|\cdot\|$ is a norm on $\mathbf{R}^m$. This problem can be interpreted as finding the best approximation of the vector b (under the norm $\|\cdot\|$) from all convex combinations of the column vectors of the matrix A, *i.e.*, projecting the vector b (with respect to the norm $\|\cdot\|$) onto the convex hull of the column vectors of A.

Linear equality and inequality constraints

The constraints we have seen so far are all special cases of linear constraints, where the corresponding optimization problem has the form

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & Cx \preceq d \\ & Gx = h, \end{array} \tag{6.12}$$

where $x \in \mathbf{R}^n$ is the variable, $C \in \mathbf{R}^{p \times n}$, $d \in \mathbf{R}^p$, $G \in \mathbf{R}^{q \times n}$ and $h \in \mathbf{R}^q$ are given data. For example, the box constrained problem (6.9) can be expressed in the form of (6.12) by defining

$$C = \begin{bmatrix} I \\ -I \end{bmatrix}, \quad d = \begin{bmatrix} u \\ -l \end{bmatrix}, \quad G = 0, \quad h = 0,$$

where I is the $n \times n$ identity matrix. We will see later that many useful constrained optimization problems can be expressed in the form of (6.12).

Geometrically, the linear inequality and equality constraints in (6.12) restrict the variable $x \in \mathbf{R}^n$ to lie in the intersection of finite hyperplanes and halfspaces, *i.e.*, a polyhedron in $\mathbf{R}^n$.

The linear constrained problem (6.12) is a convex optimization problem if the objective function f_0 is convex. In particular, if the objective function f_0 is affine (or linear, since constant terms in the objective can always be dropped), then the problem (6.12) is a linear program; if the objective function f_0 is a convex quadratic function, then the problem (6.12) is a quadratic program.

Trust region constraints

We can also constrain the variable $x \in \mathbf{R}^n$ to lie in some norm ball. For example, consider the problem

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & \|x - x_0\| \leq t, \end{array} \tag{6.13}$$

where $x \in \mathbf{R}^n$ is the variable, $x_0 \in \mathbf{R}^n$ is a given point, $t > 0$ is a given radius, and $\|\cdot\|$ is a norm on $\mathbf{R}^n$. The constraint in (6.13) requires that the variable x must lie in the ball centered at x_0 with radius t, with respect to the norm $\|\cdot\|$. Hence, it is sometimes called a *trust region constraint*. As a special case, if we take $x_0 = 0$ and the ℓ_∞-norm, then the problem (6.13) reduces to the box constrained problem (6.10).

Trust region constraints appear in different applications. For example, in the context of model fitting and estimation, we can use a trust region constraint to restrict the solution to be close to some initial guess x_0 that represents our prior knowledge. Moreover, when the objective function f_0 in (6.13) is some convex approximation of the true (possibly nonconvex) objective function, the trust region constraint can be interpreted as restricting the solution to be close to the point x_0 where the approximation is accurate, so that the solution of the problem (6.13) can be a good approximate solution of the original problem; see also §3.3.

6.2 Underdetermined equations

Consider a system of linear equations

$$Ax = b \tag{6.14}$$

in the variable $x \in \mathbf{R}^n$, where $A \in \mathbf{R}^{m \times n}$ is *fat*, *i.e.*, $m \leq n$, and $b \in \mathbf{R}^m$. We also assume that the matrix A has full rank, *i.e.*, $\mathbf{rank}\, A = m$. If $m = n$, then the system of equations (6.14) has a unique solution given by $x = A^{-1}b$. If $m < n$, then (6.14) has infinitely many solutions. In particular, let $x_0 \in \mathbf{R}^n$ be a specific solution of the linear system, *i.e.*, $Ax_0 = b$, then the set of all solutions of the equations (6.14) is given by

$$\{x \in \mathbf{R}^n \mid Ax = b\} = \{x_0 + z \mid z \in \mathcal{N}(A)\}, \tag{6.15}$$

where $\mathcal{N}(A) = \{z \in \mathbf{R}^n \mid Az = 0\}$ is the nullspace of the matrix A. Geometrically, the set (6.15) is an affine set in $\mathbf{R}^n$ that can be obtained by translating the nullspace $\mathcal{N}(A)$ by any specific solution x_0 of the linear system (6.14).

6.2.1 Least norm problems

The *least norm problem* associated with the linear system (6.14) is the optimization problem

$$\begin{array}{ll} \text{minimize} & \|x\| \\ \text{subject to} & Ax = b \end{array} \tag{6.16}$$

with variable $x \in \mathbf{R}^n$, where $\|\cdot\|$ is a norm on $\mathbf{R}^n$. A solution of the problem (6.16) is called a *least norm solution* of the linear system (6.14), since it is a solution of the equations that has the smallest norm among all solutions. The problem (6.16) is a convex optimization problem, and it is only interesting when $m < n$, since otherwise the feasible set of the problem (6.16) is the singleton $\{A^{-1}b\}$, so the solution is trivial.

Remark 6.1 The least norm problem (6.16) can be interpreted from various perspectives.

Geometric interpretation. Since the solution set of the underdetermined system (6.14) defines an affine set, solving the least norm problem (6.16) corresponds to finding a point in this affine set that is closest to the zero vector 0, with respect to the norm $\|\cdot\|$. In other words, a least norm solution of the equations (6.14) is a projection (under the norm $\|\cdot\|$) of the zero vector onto the affine set defined by the underdetermined linear system $Ax = b$.

Estimation interpretation. From an estimation perspective, let $x \in \mathbf{R}^n$ be a parameter vector that we want to estimate, and the equations $Ax = b$ represent a series of perfect (*i.e.*, noiseless) linear measurements. Note that since $m < n$, we have less measurements than the number of parameters, so the equations $Ax = b$ are not sufficient to determine a unique parameter vector x. In other words, any parameter vector x that satisfies the equations $Ax = b$ is consistent with the measurements, so we should use our prior knowledge about x to select a specific solution of the equations $Ax = b$ as our estimate. In the case of the least norm problem (6.16), the prior knowledge is that the parameter vector x should be small in the sense of the norm $\|\cdot\|$, *i.e.*, an estimate of x with a smaller norm is more likely to be the true parameter vector than an estimate with a larger norm. (These ideas can be made more formal under a Bayesian framework; see page 204.)

The least norm problem (6.16) is closely related to the norm approximation problem (4.4) on page 106. To see this, let $x_0 \in \mathbf{R}^n$ be a specific solution of the equations (6.14), and let $Z \in \mathbf{R}^{n\times k}$ be a matrix whose columns form a basis of the nullspace $\mathcal{N}(A)$, so that every solution of the equations (6.14) can be expressed as $x = x_0 + Zy$ for some $y \in \mathbf{R}^k$. Then the least norm problem (6.16) can be reformulated as an unconstrained optimization problem

$$\text{minimize} \quad \|x_0 + Zy\|$$

with variable $y \in \mathbf{R}^k$, which is a norm approximation problem of the zero vector by the affine function $f(y) = x_0 + Zy$. Hence, many discussions about approximation problems presented in §4.1 are readily translated to the least norm problem (6.16).

Some examples illustrate these ideas.

Example 6.4 *Least squares solution of underdetermined systems.* If the (squared) ℓ_2-norm is used in the least norm problem (6.16), *i.e.*,

$$\begin{array}{ll} \text{minimize} & \|x\|_2^2 \\ \text{subject to} & Ax = b, \end{array} \tag{6.17}$$

then the solution is called the *least squares solution* of the underdetermined system (6.14). Similar to the least squares approximation problem (4.6), the least norm problem (6.17) has an analytical solution given by

$$x^\star = A^T(AA^T)^{-1}b,$$

where we have used the assumption that A has full rank, so AA^T is invertible. To see that this $x^\star$ is indeed the solution of the problem (6.17), we first verify that $x^\star$ is a solution of the equations $Ax = b$: By substituting $x^\star$ into the left-hand side of the equations $Ax = b$, we have

$$Ax^\star = AA^T(AA^T)^{-1}b = b.$$

Then we verify that $x^\star$ has the smallest ℓ_2-norm among all solutions of the equations $Ax = b$. Suppose that x is an arbitrary solution of the equations $Ax = b$, then we have $A(x - x^\star) = 0$, and hence

$$\begin{array}{rcl} (x - x^\star)^T x^\star & = & (x - x^\star)^T A^T(AA^T)^{-1}b \\ & = & (A(x - x^\star))^T(AA^T)^{-1}b \\ & = & 0. \end{array}$$

This shows that $(x - x^\star) \perp x^\star$, therefore we have

$$\|x\|_2^2 = \|x - x^\star + x^\star\|_2^2 = \|x - x^\star\|_2^2 + \|x^\star\|_2^2 \geq \|x^\star\|_2^2,$$

i.e., $x^\star$ has the smallest ℓ_2-norm among all solutions of the equations $Ax = b$.

Geometrically, the least squares solution $x^\star$ of the underdetermined system (6.14) is the Euclidean projection of the zero vector onto the affine set defined by the equations $Ax = b$. This is illustrated in figure 6.1.

Example 6.5 *Least ℓ_1-norm problem and sparse solutions.* If the ℓ_1-norm is used in the least norm problem (6.16), then we have the *least ℓ_1-norm problem*

$$\begin{array}{ll} \text{minimize} & \|x\|_1 \\ \text{subject to} & Ax = b, \end{array} \tag{6.18}$$

where $x \in \mathbf{R}^n$ is the variable. According to the discussions in §4.1.3, we know that the problem (6.18) tends to promote *sparse* solutions, *i.e.*, tends to result in a solution of the underdetermined linear system $Ax = b$ that has as many zero components as possible. (See also example 6.6 for more discussions about finding sparse solutions of underdetermined equations.)

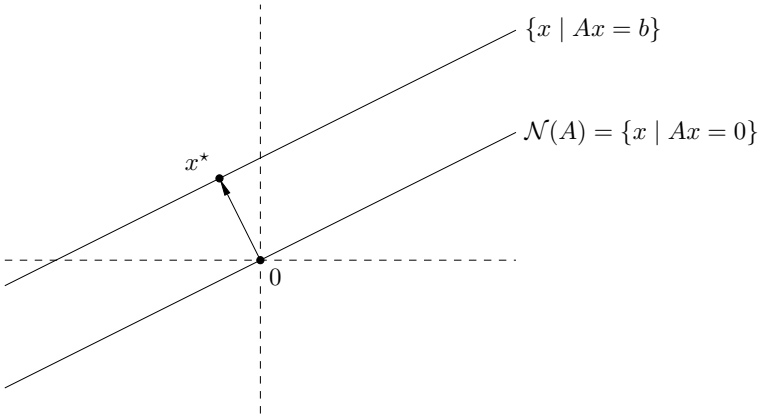

Figure 6.1 Geometric interpretation of the problem (6.17). The solution $x^\star$ of the problem (6.17) is the Euclidean projection of the zero vector onto the affine set $\{x \mid Ax = b\}$.

6.2.2 Least penalty problems

We can generalize the norm in the objective of the problem (6.16) to a broader class of penalty functions, which leads to the *least penalty problem* of the form

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^n \phi(x_i) \\ \text{subject to} & Ax = b, \end{array} \tag{6.19}$$

where $x \in \mathbf{R}^n$ is the variable, $A \in \mathbf{R}^{m\times n}$ and $b \in \mathbf{R}^m$ are given data, and $\phi\colon \mathbf{R} \to \mathbf{R}$ is a penalty function applied to each component of x.

Usually, the penalty function ϕ is chosen to be convex, nonnegative, and satisfies $\phi(0) = 0$, so that the least penalty problem (6.19) is a convex optimization problem. Similar to the class of penalty function approximation problems discussed in §4.1.3, here, by choosing different penalty functions, we can induce different characteristics in the solution of the problem (6.19).

Some nonconvex penalty functions for the problem (6.19) are also useful in practice. An example is presented below.

Example 6.6 *Least cardinality problems.* Consider the least penalty problem (6.19) with the penalty function ϕ being the *cardinality function*, which counts the number of nonzero entries of a vector. This leads to the *least cardinality problem*, given by

$$\begin{array}{ll} \text{minimize} & \mathbf{card}\, x \\ \text{subject to} & Ax = b, \end{array} \tag{6.20}$$

where the variable is $x \in \mathbf{R}^n$. The least cardinality problem (6.20) consists in finding a solution of the underdetermined linear system $Ax = b$ that has as few nonzero components as possible, *i.e.*, a sparsest solution of the equations $Ax = b$.

The problem (6.20) is not a convex optimization problem, since the cardinality function is not convex. However, if the dimension n is not too large, we can solve the

problem via combinatorial search. Specifically, suppose that we want to find a solution of the equations $Ax = b$ with k nonzero components, where $k \leq n$ is a given positive integer, then we can enumerate all possible choices of the k indices out of $1, \ldots, n$. For each choice of the k indices, the equations $Ax = b$ reduce to $\tilde{A}\tilde{x} = b$, where $\tilde{A}$ is the submatrix of A formed by the columns corresponding to the chosen indices, and $\tilde{x}$ is the subvector of x formed by the components corresponding to the chosen indices. If for some choice of the k indices, the matrix $\tilde{A}$ is nonsingular, then $\tilde{x} = \tilde{A}^{-1}b$ is a solution of the equations $Ax = b$ with k nonzero components. If for some value of k, all possible choices of the indices lead to a singular matrix $\tilde{A}$ (and $b \notin \mathcal{R}(\tilde{A})$), then there is no solution of the equations $Ax = b$ with this number of nonzero components. By repeating this procedure for $k = 1, \ldots, n$, we can then solve the least cardinality problem (6.20).

Solving the least cardinality problem (6.20) via combinatorial search is not practically feasible when the dimension n gets larger, since for each $k = 1, \ldots, n$, we need to check $n!/(k!(n-k)!)$ possible choices of the indices (in the worst case). Here, again, the ℓ_1-norm heuristic given by the problem (6.18) often works quite well as a convex surrogate of (6.20), for finding a sparse solution of the underdetermined linear system $Ax = b$.

Bayesian interpretation

We can interpret the least penalty problem (6.19) from a Bayesian perspective. Suppose that we want to estimate the value of a random variable $x \in \mathbf{R}^n$ based on some perfect linear measurements $Ax = b$, where $A \in \mathbf{R}^{m\times n}$ and $b \in \mathbf{R}^m$. Then the likelihood function of x under the observations b (*i.e.*, the posterior distribution of b given x) is expressed as

$$p_{b|x}(x, b) = \begin{cases} 1, & Ax = b \\ 0, & \text{otherwise}, \end{cases}$$

so the log-likelihood function of the variable x is

$$\log p_{b|x}(x, b) = \begin{cases} 0, & Ax = b \\ -\infty, & \text{otherwise}. \end{cases}$$

Assume that the components x_i, $i = 1, \ldots, n$, of the variable $x \in \mathbf{R}^n$ are IID under the prior density $p\colon \mathbf{R} \to \mathbf{R}_+$ defined as

$$p(x_i) = \frac{\exp(-\phi(x_i))}{\int \exp(-\phi(u))\ du},$$

where $\phi\colon \mathbf{R} \to \mathbf{R}$ is a penalty function corresponding to the problem (6.19), then the log-prior distribution of x is given by

$$\log p_x(x) = \log \prod_{i=1}^n p(x_i) = -\sum_{i=1}^n \phi(x_i) - n \log \left(\int \exp(-\phi(u))\ du \right).$$

As a result, the maximum a posteriori estimation objective of the variable x is expressed as

$$\log p_{b|x}(x,b) + \log p_x(x) = \begin{cases} -\sum_{i=1}^n \phi(x_i), & Ax = b \\ -\infty, & \text{otherwise.} \end{cases}$$

(Note that we have dropped the constant term in the log-prior of x.) Hence, the maximum a posteriori estimation of x under the observations b is formulated as the optimization problem

$$\begin{array}{ll} \text{maximize} & -\sum_{i=1}^n \phi(x_i) \\ \text{subject to} & Ax = b, \end{array}$$

which is the least penalty problem (6.19) by negating the objective.

Conversely, a maximum a posteriori estimation problem of the variable $x \in \mathbf{R}^n$ under the perfect linear measurements $Ax = b$ and IID components x_i with prior density $p\colon \mathbf{R} \to \mathbf{R}_+$, given by

$$\begin{array}{ll} \text{maximize} & \sum_{i=1}^n \log p(x_i) \\ \text{subject to} & Ax = b, \end{array}$$

can be interpreted as a least penalty problem of the form (6.19), where the penalty function $\phi\colon \mathbf{R} \to \mathbf{R}$ is defined as $\phi(u) = -\log p(u)$ for $u \in \mathbf{R}$.

6.3 Probabilities and distributions

Consider a nonparametric distribution estimation problem where we want to estimate a probability distribution for some discrete random variable $Z \in \{c_1, \ldots, c_n\}$ with n possible outcomes. For simplicity of presentation, here we assume that the random variable Z is scalar, *i.e.*, $c_i \in \mathbf{R}$ for all $i = 1, \ldots, n$, but the same ideas readily generalize to the case where Z takes values in higher dimensional spaces or some abstract sets.

Let $x \in \mathbf{R}^n$ be a vector variable representing the distribution of Z, *i.e.*, $x_i = \mathbf{prob}(Z = c_i)$ for $i = 1, \ldots, n$. Then the most fundamental requirements for the variable x are given by the probability constraints

$$x \succeq 0 \quad \text{and} \quad \mathbf{1}^T x = 1, \tag{6.21}$$

i.e., the variable x must lie in the probability simplex in $\mathbf{R}^n$. A nonparametric distribution estimation problem with prior information about the distribution of Z can be formulated as the optimization problem

$$\begin{array}{lll} \text{minimize} & f_0(x) & \\ \text{subject to} & x \succeq 0, \quad \mathbf{1}^T x = 1, \\ & x \in \mathcal{X}, \end{array} \tag{6.22}$$

where $f_0\colon \mathbf{R}^n \to \mathbf{R}$ is some objective function (*e.g.*, as presented in §4.3) and $\mathcal{X} \subseteq \mathbf{R}^n$ is some constraint set representing the additional prior information about

the distribution of Z. In the next few paragraphs, we will see that many types of the prior information, *i.e.*, the constraint set $\mathcal{X}$, can be represented via convex inequality and equality constraints, so that the problem (6.22) is a convex program (assuming that the objective function f_0 is convex).

Moment constraints

The kth *moment* (about zero) of the random variable $Z \in \{c_1, \ldots, c_n\}$ is defined as the expectation of the kth power of Z, *i.e.*,

$$\mathbf{E}\, Z^k = \sum_{i=1}^n c_i^k x_i, \tag{6.23}$$

where $x \in \mathbf{R}^n$ satisfies the probability constraints (6.21) representing the distribution of Z. As a special case, when $k = 1$, the first moment $\mathbf{E}\, Z$ reduces to the *mean* of the random variable Z. According to (6.23), the kth moment of the random variable Z is a linear function of the distribution $x \in \mathbf{R}^n$ for all $k \in \mathbf{Z}_{++}$. Hence, it is easily seen that if we have some prior information about $\mathbf{E}\, Z^k$, such as its value, upper or lower bounds, etc., then we can represent such knowledge via linear equality or inequality constraints on the distribution x.

Example 6.7 *Moment constraints.* Suppose that we know the mean of the random variable Z is equal to some value $\mu \in \mathbf{R}$, then we can represent this information via the linear equality constraint

$$\mathbf{E}\, Z = c^T x = \mu$$

with variable $x \in \mathbf{R}^n$ being the distribution of Z, where $c = (c_1, \ldots, c_n) \in \mathbf{R}^n$ is the vector of possible outcomes of Z. This is readily generalized to the kth moment of Z. For example, if we know that the kth moment of Z is equal to some value $\eta \in \mathbf{R}$, then by (6.23), we have

$$\mathbf{E}\, Z^k = \sum_{i=1}^n c_i^k x_i = \eta,$$

which is also a linear equality constraint on the distribution x.

Moreover, if we are given some bounds on the kth moment of Z, *e.g.*,

$$\mathbf{E}\, Z^k \in [\alpha, \beta],$$

where $\alpha, \beta \in \mathbf{R}$ are given scalars, then we can represent this information via the linear inequality constraints

$$\alpha \leq \sum_{i=1}^n c_i^k x_i \leq \beta$$

with variable $x \in \mathbf{R}^n$.

The idea of moment constraints can be extended to restricting the expectation of some general function of the random variable Z. Specifically, let $g\colon \mathbf{R} \to \mathbf{R}$ be a function with domain $\mathbf{dom}\, g$ satisfying $\{c_1, \ldots, c_n\} \subseteq \mathbf{dom}\, g$, then the expectation of $g(Z)$ is given by

$$\mathbf{E}\, g(Z) = \sum_{i=1}^n g(c_i) x_i, \tag{6.24}$$

which is a linear function of the distribution $x \in \mathbf{R}^n$. When the function g has the form of the power function $g(u) = u^k$ for some positive integer $k \in \mathbf{Z}_{++}$, the expectation $\mathbf{E}\, g(Z)$ reduces to the kth moment of Z.

Probability constraints

As a special case of the function expectation constraints given by (6.24), let $S \subseteq \mathbf{R}$ be a subset of the real numbers such that $S \cap \{c_1, \ldots, c_n\} \neq \emptyset$. Then the probability of the random variable $Z \in \{c_1, \ldots, c_n\}$ taking values in the set S is given by the following linear function of the distribution $x \in \mathbf{R}^n$:

$$\mathbf{prob}(Z \in S) = p^T x,$$

where the vector $p \in \mathbf{R}^n$ is defined as

$$p_i = \begin{cases} 1, & c_i \in S \\ 0, & \text{otherwise} \end{cases} \tag{6.25}$$

for $i = 1, \ldots, n$. Therefore, known probabilities or probability bounds of certain events related to the random variable Z can be represented via linear equality or inequality constraints on the distribution $x \in \mathbf{R}^n$. For example, suppose we have the prior information that the probability of Z taking nonnegative values is at least $\mu \in [0, 1]$, then we can represent this knowledge via the linear inequality constraint

$$\mathbf{prob}(Z \geq 0) = p^T x \geq \mu,$$

where the vector $p \in \mathbf{R}^n$ is defined similarly as in (6.25), with $S = \mathbf{R}_+$.

The idea of probability constraints can be extended to conditional probabilities. The following example illustrates this idea.

Example 6.8 *Conditional probability constraints.* Let $A, B \subseteq \mathbf{R}$ be two subsets of the real numbers, then the conditional probability of $Z \in A$ given $Z \in B$ is expressed as

$$\mathbf{prob}(Z \in A \mid Z \in B) = \frac{\mathbf{prob}(Z \in A \cap B)}{\mathbf{prob}(Z \in B)} = \frac{p^T x}{q^T x}$$

(assuming that all probabilities are nonzero), where each component of the vectors $p, q \in \mathbf{R}^n$ is given by

$$p_i = \begin{cases} 1, & c_i \in A \cap B \\ 0, & \text{otherwise} \end{cases} \quad \text{and} \quad q_i = \begin{cases} 1, & c_i \in B \\ 0, & \text{otherwise.} \end{cases}$$

Suppose we know that the conditional probability $\mathbf{prob}(Z \in A \mid Z \in B)$ is bounded in the interval $[\alpha, \beta]$ for some $\alpha, \beta \in [0, 1]$, then this knowledge can be represented as

$$\alpha \leq \frac{p^T x}{q^T x} \leq \beta.$$

Since $q^T x > 0$, this is equivalent to the linear inequality constraints

$$p^T x \geq \alpha q^T x \quad \text{and} \quad p^T x \leq \beta q^T x,$$

i.e.,

$$(p - \alpha q)^T x \geq 0 \quad \text{and} \quad (\beta q - p)^T x \geq 0,$$

on the distribution $x \in \mathbf{R}^n$.

Variance constraints

Some prior information about the distribution of Z can be represented via nonlinear convex constraints. As a basic example, the *variance* of the random variable $Z \in \{c_1, \ldots, c_n\}$ with distribution $x \in \mathbf{R}^n$ satisfying (6.21) is defined as

$$\mathbf{var}\, Z = \mathbf{E}\,(Z - \mathbf{E}\, Z)^2 = \mathbf{E}\, Z^2 - (\mathbf{E}\, Z)^2 = \sum_{i=1}^n c_i^2 x_i - \left(\sum_{i=1}^n c_i x_i\right)^2. \tag{6.26}$$

Noticing that the first term in the right-hand side of (6.26) is linear in x and the second term can be expressed as

$$\left(\sum_{i=1}^n c_i x_i\right)^2 = \sum_{i=1}^n \sum_{j=1}^n c_i c_j x_i x_j = x^T (cc^T) x,$$

which is a convex quadratic function of the distribution $x \in \mathbf{R}^n$, we see that the variance $\mathbf{var}\, Z$ is concave in x. Therefore, a *lower bound* on the variance of Z, say, $\mathbf{var}\, Z \geq \mu$ for some $\mu \in \mathbf{R}$, can be represented via a convex inequality constraint

$$x^T (cc^T) x - \sum_{i=1}^n c_i^2 x_i \leq -\mu$$

with variable $x \in \mathbf{R}^n$.

Remark 6.2 *Variance upper bounds.* An *upper bound* on the variance of Z, given by

$$\mathbf{var}\, Z = \sum_{i=1}^n c_i^2 x_i - x^T (cc^T) x \leq \eta$$

for some $\eta \in \mathbf{R}$, is *not* a convex constraint. However, it is a difference-of-convex constraint, since the variance $\mathbf{var}\, Z$ is the difference of a linear function and a convex quadratic function of x. Therefore, if the objective f_0 is a convex (or difference-of-convex) function, the distribution estimation problem in the form (6.22) with a variance upper bound constraint, given by

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & x \succeq 0, \quad \mathbf{1}^T x = 1, \\ & \sum_{i=1}^n c_i^2 x_i - x^T (cc^T) x \leq \eta, \end{array}$$

is a difference-of-convex program, which can be approximately solved by the convex-concave procedure presented in §3.2.3.

Entropy constraints

Another type of nonlinear convex constraints that are useful in distribution estimation problems of the form (6.22) are the *entropy* constraints. Let $x \in \mathbf{R}^n$ with $x \succeq 0$

and $\mathbf{1}^T x = 1$ be a probability distribution of the random variable $Z \in \{c_1, \ldots, c_n\}$. The *negative entropy function* of the distribution x is defined as

$$\sum_{i=1}^{n} x_i \log x_i,$$

which is a convex function of x. If we have the prior information that the *entropy* of the distribution x is *bounded below* by some value $\mu \in \mathbf{R}$, *i.e.*, the negative entropy of x is bounded above by $-\mu$, then this prior information can be represented via the convex inequality constraint

$$\sum_{i=1}^{n} x_i \log x_i \leq -\mu.$$

Intuitively, this constraint restricts the distribution x to have a certain level of *uncertainty*, since a higher entropy corresponds to a more uncertain distribution. In other words, it requires that the distribution x to be sufficiently *spread out* over the possible outcomes of Z, rather than being too concentrated on a few outcomes. (See also page 134 and exercise 4.6.)

Moreover, if we are given some reference distribution $q \in \mathbf{R}^n$ such that $q \succeq 0$ and $\mathbf{1}^T q = 1$, the *relative entropy* (or *KL-divergence*) between the two distributions x and q is defined as

$$\sum_{i=1}^{n} x_i \log \frac{x_i}{q_i},$$

which is also a convex function of x. Then it follows that we can restrict the *maximum* divergence between x and the given reference distribution q, as a convex inequality constraint on the distribution x. Such a constraint can be interpreted as requiring the distribution x to be sufficiently close to the reference distribution q.

Numerical example

We present a numerical example to demonstrate some applications of the constrains mentioned above in the nonparametric distribution estimation problem (6.22).

Suppose $Z \in \{c_1, \ldots, c_n\}$ is a discrete random variable with $n = 200$ possible outcomes, separated with equal spacing in the interval $[-0.5, 1]$. The constraint set $\mathcal{X} \subseteq \mathbf{R}^n$ in the distribution estimation problem (6.22) representing our prior information is given by the following inequality constraints:

$$\begin{array}{rcl} \mathbf{E}\, Z & \in & [-0.2, 0.5] \\ \mathbf{E}\, Z^2 & \leq & 0.2 \\ \mathbf{E}(3Z^4 - 7Z^3) & \leq & -0.7 \\ \mathbf{var}\, Z & \geq & 0.1 \\ \mathbf{prob}(Z < 0) & \geq & 0.35. \end{array} \tag{6.27}$$

Let $x \in \mathbf{R}^n$ be the variable representing the distribution of the random variable Z, to be estimated. According to the previous discussions, the first three expectation and

the last probability constraints can all be represented as linear inequality constraints on x, while the fourth variance constraint is a convex quadratic inequality constraint on x. Therefore, the constraint set $\mathcal{X}$ in (6.22) obtained from the intersection of these inequality constraints is a convex set.

First consider the problem of finding a distribution $x \in \mathbf{R}^n$ of the random variable Z that has the maximum entropy among all distributions satisfying the constraints in (6.27). This can be formulated as the optimization problem

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^n x_i \log x_i \\ \text{subject to} & x \succeq 0, \quad \mathbf{1}^T x = 1, \\ & -0.2 \leq \mathbf{E}\, Z \leq 0.5, \quad \mathbf{E}\, Z^2 \leq 0.2, \\ & \mathbf{E}(3Z^4 - 7Z^3) \leq -0.7, \quad \mathbf{var}\, Z \geq 0.1, \\ & \mathbf{prob}(Z < 0) \geq 0.35, \end{array} \tag{6.28}$$

where the variable is $x \in \mathbf{R}^n$. Since the objective function is convex and the constraints are all convex inequality and equality constraints, the maximum entropy estimation problem (6.28) is a convex optimization problem. Figure 6.2 shows the optimal distribution $x^\star \in \mathbf{R}^n$ of the problem (6.28). The maximum entropy distribution $x^\star$ satisfies

$$\begin{aligned} \mathbf{E}\, Z &= 0.22 \\ \mathbf{E}\, Z^2 &= 0.2 \\ \mathbf{E}(3Z^4 - 7Z^3) &= -0.7 \\ \mathbf{var}\, Z &= 0.15 \\ \mathbf{prob}(Z < 0) &= 0.35, \end{aligned}$$

i.e., except for the mean and variance constraints, all the other constraints in (6.27) are *tight* at the optimal distribution $x^\star$.

Now suppose among all distributions of Z satisfying the constraints in (6.27), we would like to compute the upper and lower bounds on the probability

$$\mathbf{prob}(Z \leq c_i) = p^T x \tag{6.29}$$

for all $i = 1, \ldots, n$, where the indicator vector $p \in \mathbf{R}^n$ is given by

$$p_j = \begin{cases} 1, & c_j \leq c_i \\ 0, & \text{otherwise} \end{cases} \tag{6.30}$$

for all $j = 1, \ldots, n$. Noticing that since $c_1 < c_2 < \cdots < c_n$, the probability (6.29) is actually the cumulative distribution of the random variable Z at the point c_i. To compute the pointwise lower bound on the probability (6.29) for each $i = 1, \ldots, n$, we can solve the following optimization problem:

$$\begin{array}{ll} \text{minimize} & \mathbf{prob}(Z \leq c_i) \\ \text{subject to} & x \succeq 0, \quad \mathbf{1}^T x = 1, \\ & -0.2 \leq \mathbf{E}\, Z \leq 0.5, \quad \mathbf{E}\, Z^2 \leq 0.2, \\ & \mathbf{E}(3Z^4 - 7Z^3) \leq -0.7, \quad \mathbf{var}\, Z \geq 0.1, \\ & \mathbf{prob}(Z < 0) \geq 0.35, \end{array} \tag{6.31}$$

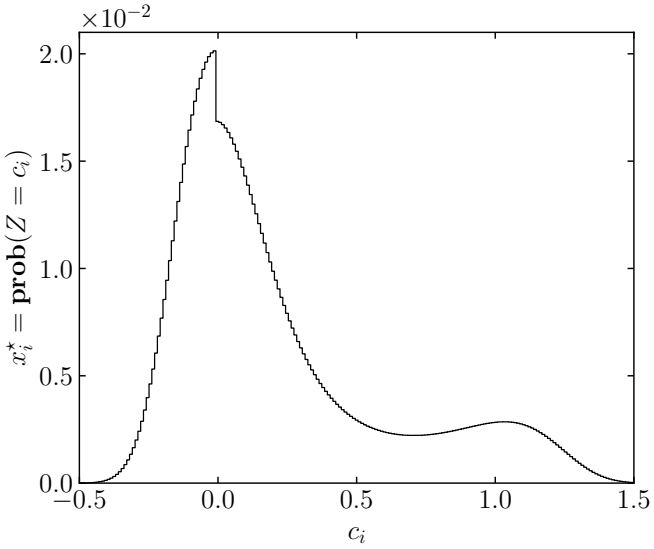

Figure 6.2 The maximum entropy distribution $x^\star \in \mathbf{R}^n$ that satisfies the constraints in (6.27), obtained by solving the problem (6.28).

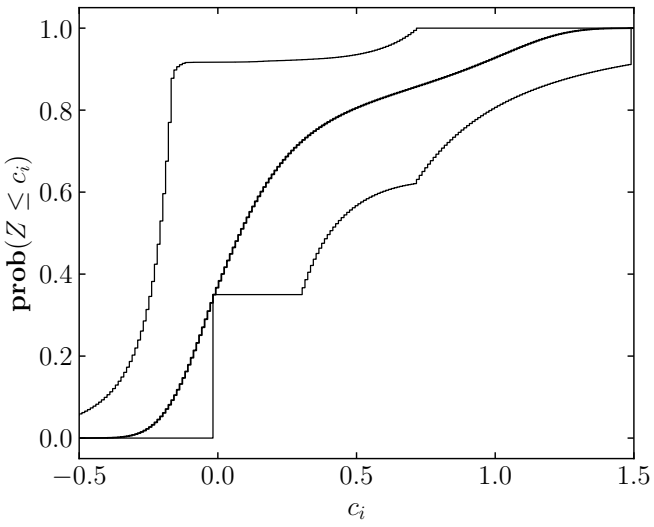

Figure 6.3 The top and bottom curves plot the upper and lower bounds on the cumulative distribution of Z at each point c_i, $i = 1, \ldots, n$, among all distributions satisfying the constraints in (6.27). The cumulative distribution of the maximum entropy distribution $x^\star$ from the problem (6.28) is shown thicker in the middle.

where $x \in \mathbf{R}^n$ is the variable. By (6.29) and (6.30), the objective function in the problem (6.31) is a linear function of x, so the problem (6.31) is a convex optimization problem. To compute the upper bound, we only need to change the objective in the problem (6.31) to maximize the probability $\mathbf{prob}(Z \leq c_i)$. Figure 6.3 shows the pointwise upper (the top curve) and lower (the bottom curve) bounds on the cumulative distribution of Z at each point c_i, $i = 1, \ldots, n$. The cumulative distribution of the maximum entropy distribution $x^\star$ from the problem (6.28) at each point c_i is shown thicker in the middle.

6.4 Functional constraints

6.4.1 Function fitting problems

A special subclass of the approximation problems is to find a function that best *fits* or *interpolates* some given data points. Specifically, let

$$(z_i, y_i) \in \mathbf{R}^k \times \mathbf{R}, \quad i = 1, \ldots, m,$$

be the given data points, and let $f : \mathbf{R}^k \to \mathbf{R}$ with

$$f(z) = x_1 f_1(z) + \cdots + x_n f_n(z) \tag{6.32}$$

be a linear combination of a group of given *basis functions* $f_1, \ldots, f_n : \mathbf{R}^k \to \mathbf{R}$ with variable coefficients $x_1, \ldots, x_n \in \mathbf{R}$. Then the problem of *function fitting* on the dataset (z_i, y_i), $i = 1, \ldots, m$, can be formulated as the optimization problem

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^m \phi(f(z_i) - y_i) \\ \text{subject to} & x \in \mathcal{X} \end{array} \tag{6.33}$$

with variable $x \in \mathbf{R}^n$, where the function $f : \mathbf{R}^k \to \mathbf{R}$ in the objective is given by (6.32), and $\phi : \mathbf{R} \to \mathbf{R}$ is a penalty function that measures the approximation error of the function f on the dataset. We will see later that the constraint set $\mathcal{X} \subseteq \mathbf{R}^n$ can be used to represent some prior information about the structure or properties of the target function f.

Remark 6.3 *Function fitting as penalty function approximation.* The function fitting problem (6.33) can be expressed in the form of the penalty function approximation problem (4.12) on page 111. To do this, for all $i = 1, \ldots, m$, we define

$$a_i = (f_1(z_i), \ldots, f_n(z_i)) \in \mathbf{R}^n,$$

so that $f(z_i) = a_i^T x$ for all $i = 1, \ldots, m$. Since the basis functions $f_1, \ldots, f_n$ are given, the vectors $a_1, \ldots, a_m$ are also fixed and known given the dataset (z_i, y_i), $i = 1, \ldots, m$. Then the function fitting problem (6.33) can be rewritten as

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^m \phi(a_i^T x - y_i) \\ \text{subject to} & x \in \mathcal{X} \end{array} \tag{6.34}$$

with variable $x \in \mathbf{R}^n$, which is essentially a (constrained) penalty function approximation problem corresponding to the system of linear equations $a_i^T x = y_i$ for $i = 1, \ldots, m$.

According to the formulation (6.34), it follows that the function fitting problem (6.33) is a convex optimization problem if the penalty function ϕ is convex and the constraint set $\mathcal{X}$ is a convex set.

Example 6.9 *Polynomial fitting.* Let $z_i \in \mathbf{R}$ and $y_i \in \mathbf{R}$ be the given data points for $i = 1, \ldots, m$, and let

$$f(z) = x_1 + x_2 z + \cdots + x_n z^{n-1}$$

be a polynomial function of degree at most $n-1$ with variable coefficients $x \in \mathbf{R}^n$. Then the problem of fitting this polynomial function to the dataset (z_i, y_i), $i = 1, \ldots, m$, can be expressed in the form of (6.33) as

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^m \left(\sum_{j=1}^n x_j z_i^{j-1} - y_i \right)^2 \\ \text{subject to} & x \in \mathcal{X} \end{array} \tag{6.35}$$

with variable $x \in \mathbf{R}^n$. Here, as a basic example, the penalty function $\phi \colon \mathbf{R} \to \mathbf{R}$ in (6.33) is chosen as the quadratic penalty $\phi(u) = u^2$ for $u \in \mathbf{R}$, so the resulting problem (6.35) is a constrained least squares problem.

6.4.2 Constraints

In this section, we present some ideas of how different constraints on the variable $x \in \mathbf{R}^n$ can be used in the function fitting problem (6.33) to impose various properties on the target function f. Here we always take the assumption that the target function $f \colon \mathbf{R}^k \to \mathbf{R}$ of the problem (6.33) is given by a linear combination of some fixed basis functions as in (6.32). However, it is worth noting that in general, the function f need not be restricted to this form; see example 3.10, page 93.

Interpolation constraints

Let $(s_i, t_i) \in \mathbf{R}^k \times \mathbf{R}$, $i = 1, \ldots, p$, be some given *interpolation points*, then the *interpolation conditions* require the target function f in the problem (6.33) to pass through all these points. By (6.32), the function f is linear in the variable $x \in \mathbf{R}^n$, so the interpolation conditions can be represented as linear equality constraints on x, *i.e.*,

$$f(s_i) = a_i^T x = t_i, \quad i = 1, \ldots, p, \tag{6.36}$$

where $a_i = (f_1(s_i), \ldots, f_n(s_i)) \in \mathbf{R}^n$.

Note that the interpolation points $(s_i, t_i) \in \mathbf{R}^k \times \mathbf{R}$, $i = 1, \ldots, p$, might overlap with the data points (z_i, y_i), $i = 1, \ldots, m$, and often in practice, the interpolation points are chosen as a subset of the data points.

When p is large, *i.e.*, there are many interpolation constraints, the function fitting problem (6.33) might be infeasible, which means that there is no function f of the form (6.32) that can pass through all the interpolation points. Specifically, let $A \in \mathbf{R}^{p \times n}$ be the matrix whose ith row is given by a_i^T in (6.36), and let $t =$

$(t_1, \ldots, t_p) \in \mathbf{R}^p$ be the vector of the target values at the interpolation points. Then the interpolation constraints (6.36) can be expressed as the linear system of equations

$$Ax = t,$$

and it follows that the interpolation constraints are feasible if and only if $t \in \mathcal{R}(A)$. A simple approach to deal with infeasible interpolation conditions is to *relax* some of the constraints by allowing approximation error, which, in other words, is equivalent to transforming some of the interpolation points into data points. (A formal treatment of relaxations and some ideas about dealing with infeasibility will be discussed in more detail in §6.5.)

Bounding constraints

We could also introduce inequality constraints on the target function f in the problem (6.33). As a basic example, we could relax the interpolation conditions by allowing the function f to be *bounded* at the points $s_i \in \mathbf{R}^k$, $i = 1, \ldots, p$, rather than being exactly equal to some target value, *i.e.*,

$$f(s_i) \in [\alpha_i, \beta_i], \quad i = 1, \ldots, p, \tag{6.37}$$

where $\alpha_i, \beta_i \in \mathbf{R}$ are given lower and upper bounds on the function values at the point s_i. Let $a_i = (f_1(s_i), \ldots, f_n(s_i)) \in \mathbf{R}^n$ for $i = 1, \ldots, p$, then the *bounding conditions* (6.37) can be expressed as the linear inequality constraints

$$\alpha_i \leq a_i^T x \leq \beta_i, \quad i = 1, \ldots, p,$$

on the variable $x \in \mathbf{R}^n$.

Example 6.10 *Bounding constraints.* Suppose we are given the data points $(z_i, y_i) \in \mathbf{R}^k \times \mathbf{R}$, $i = 1, \ldots, m$, and we would like to fit a function f of the form (6.32) to these data points. Let $a_i = (f_1(z_i), \ldots, f_n(z_i)) \in \mathbf{R}^n$ for $i = 1, \ldots, m$, then we can express the following bounding conditions as linear inequality constraints on the variable $x \in \mathbf{R}^n$.

- If the function f is required to be *nonnegative* at the points z_i for all $i = 1, \ldots, m$, then we have
$$f(z_i) = a_i^T x \geq 0, \quad i = 1, \ldots, m. \tag{6.38}$$
- Suppose that the data points satisfies $z_1 \preceq \cdots \preceq z_m$, and the function f is required to be *monotone nondecreasing* at the points z_i for all $i = 1, \ldots, m-1$, then we have
$$f(z_{i+1}) - f(z_i) = (a_{i+1} - a_i)^T x \geq 0, \quad i = 1, \ldots, m-1. \tag{6.39}$$
- If the function f is required to be a *lower bound* of the data points, then we have
$$f(z_i) = a_i^T x \leq y_i, \quad i = 1, \ldots, m.$$

We should note that the first two types of constraints mentioned above, *i.e.*, (6.38) and (6.39), do not necessarily require the function f to satisfy the corresponding properties

at all points in its domain, unless f is assumed to have some special structure. For example, suppose the target function f has domain $\mathbf{dom}\, f = \mathbf{conv}\{z_1, \ldots, z_m\}$, and is assumed to take the form of a piecewise affine function with breakpoints at z_i, $i = 1, \ldots, m$, then the nonnegativity constraints (6.38) are sufficient to guarantee $f(z) \geq 0$ for all $z \in \mathbf{dom}\, f$. In this case, the monotonicity constraints (6.39) are also sufficient to guarantee that f is monotone nondecreasing on $\mathbf{dom}\, f$.

There are several other types of bounding constraints that can be represented as linear inequality constraints on the variable $x \in \mathbf{R}^n$. For example, if we require the function f to satisfy the *Lipschitz continuity* condition at the points s_i for all $i = 1, \ldots, p$, with a given Lipschitz constant $L \in \mathbf{R}_+$, then we have

$$|f(s_i) - f(s_j)| \leq L\|s_i - s_j\|, \quad i, j = 1, \ldots, p,$$

which is equivalent to the set of linear inequality constraints

$$-L\|s_i - s_j\| \leq (a_i - a_j)^T x \leq L\|s_i - s_j\|, \quad i, j = 1, \ldots, p,$$

on the variable $x \in \mathbf{R}^n$, where $a_i = (f_1(s_i), \ldots, f_n(s_i)) \in \mathbf{R}^n$ for $i = 1, \ldots, p$.

Derivative constraints

Suppose the basis functions $f_1, \ldots, f_n \colon \mathbf{R}^k \to \mathbf{R}$ in (6.32) are differentiable at the point $s \in \mathbf{R}^k$, then the target function f given by (6.32) is also differentiable at s. In particular, the gradient of f at s is given by

$$\nabla f(s) = x_1 \nabla f_1(s) + \cdots + x_n \nabla f_n(s),$$

which is a linear function of the variable $x \in \mathbf{R}^n$. Therefore, we can impose interpolation or bounding conditions on the gradient ∇f at some point $s \in \mathbf{R}^k$ by linear equality or inequality constraints of x. Moreover, we can bound the norm of the gradient $\nabla f(s)$ as

$$\|\nabla f(s)\| \leq M$$

for some $M \in \mathbf{R}_+$. Let $A \in \mathbf{R}^{k \times n}$ be the matrix whose ith column is given by $\nabla f_i(s)$ for $i = 1, \ldots, n$, then the above gradient constraint can be expressed as

$$\|Ax\| \leq M,$$

which is a convex constraint on the variable $x \in \mathbf{R}^n$.

These ideas can be extended to higher derivatives of the target function f. For example, suppose the basis functions $f_1, \ldots, f_n \colon \mathbf{R}^k \to \mathbf{R}$ in (6.32) are twice differentiable at the point $s \in \mathbf{R}^k$, then the Hessian of f at s is given by

$$\nabla^2 f(s) = x_1 \nabla^2 f_1(s) + \cdots + x_n \nabla^2 f_n(s),$$

which is a linear function of the variable $x \in \mathbf{R}^n$. Therefore, the bounding condition

$$\alpha I \preceq \nabla^2 f(s) \preceq \beta I$$

at the point s with some $\alpha, \beta \in \mathbf{R}$ is a group of linear matrix inequality constraints on the variable $x \in \mathbf{R}^n$. As a special case, the linear matrix inequality constraint $\nabla^2 f(s) \succeq 0$ requires the target function f to have nonnegative curvature in the neighborhood of some point $s \in \mathbf{R}^k$, *i.e.*, the function f is (locally) convex at the point s. If the function f has some special structure, then this condition is sufficient to guarantee the global convexity of f. An example is presented below.

Example 6.11 *Convex constraint of quadratic functions.* Suppose the basis functions $f_1, \ldots, f_n \colon \mathbf{R}^k \to \mathbf{R}$ in (6.32) are given by the quadratic functions

$$f_i(z) = \frac{1}{2} z^T P_i z + q_i^T z + r_i, \quad i = 1, \ldots, n,$$

where $P_i \in \mathbf{S}^k$, $q_i \in \mathbf{R}^k$, and $r_i \in \mathbf{R}$ for $i = 1, \ldots, n$. Then the target function f given by (6.32) is also a quadratic function of the form

$$f(z) = \frac{1}{2} z^T P z + q^T z + r,$$

where

$$P = \sum_{i=1}^n x_i P_i, \quad q = \sum_{i=1}^n x_i q_i, \quad r = \sum_{i=1}^n x_i r_i.$$

If we require the target function f to be a convex quadratic function, then we have

$$\nabla^2 f(z) = P = \sum_{i=1}^n x_i P_i \succeq 0,$$

which is a linear matrix inequality constraint on the variable $x \in \mathbf{R}^n$.

Numerical example

We present a numerical example to illustrate the ideas presented above. Suppose we are given the data points $(z_i, y_i) \in \mathbf{R} \times \mathbf{R}$, $i = 1, \ldots, m$, with $m = 80$, which are shown as circles in figure 6.4. We consider the problem of fitting a polynomial function of degree at most $n - 1$, given by

$$f(z) = x_1 + x_2 z + \cdots + x_n z^{n-1}, \tag{6.40}$$

to these data points, and here we choose $n = 5$.

First consider the case where there is no constraint on the variable $x \in \mathbf{R}^n$. Taking the least squares penalty function $\phi(u) = u^2$ for measuring the approximation error, the resulting function fitting problem is then given by

$$\text{minimize} \quad \sum_{i=1}^m \phi(f(z_i) - y_i) = \sum_{i=1}^m \left(\sum_{j=1}^n x_j z_i^{j-1} - y_i \right)^2. \tag{6.41}$$

The polynomial function fitting result on the given dataset from solving the problem (6.41) is shown as the thicker solid curve in figure 6.4.

Now suppose we have the prior information that the target function f should satisfy the following interpolation conditions:

$$f(-5) = -0.4, \quad f(0) = 0, \quad f(8.7) = 0.1, \tag{6.42}$$

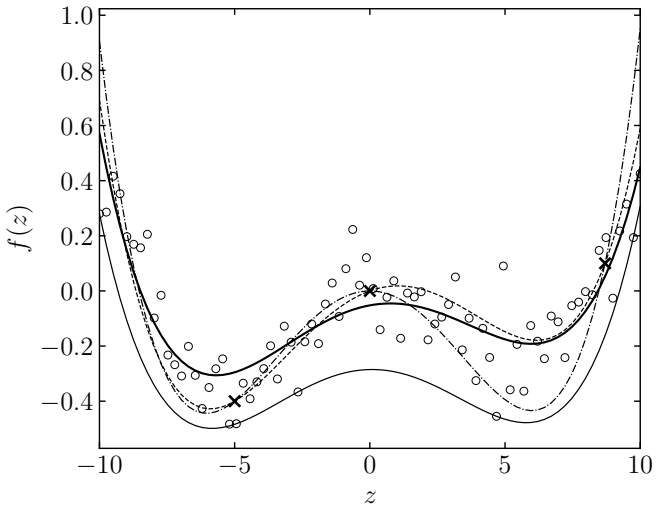

Figure 6.4 *Polynomial function fitting.* The circles plot the given data points (z_i, y_i), $i = 1, \ldots, m$. The thicker solid curve shows the unconstrained polynomial function fitting result from solving the problem (6.41). The dashed curve shows the constrained fitting by solving (6.41) under the constraints (6.42). The dashdotted curve shows the results from solving (6.41) under both constraints (6.42) and (6.43). The thinner solid curve on the bottom corresponds to a lower bounding polynomial function in the form (6.40) of the data points.

which are linear equality constraints on the variable $x \in \mathbf{R}^n$. The polynomial function fitting result from solving the problem (6.41) with the interpolation constraints (6.42) is shown as the dashed curve in figure 6.4, where the interpolation points are shown as crosses.

We could further introduce prior information about the gradient of the target function f at some points. Specifically, we require f to have zero gradient at the point $z = 0$, and the gradient of f at $z = 2.5$ to be bounded above by -0.1, *i.e.*,

$$\nabla f(0) = 0, \quad \nabla f(2.5) \leq -0.1. \tag{6.43}$$

By (6.40), we have

$$\nabla f(z) = x_2 + 2x_3 z + \cdots + (n-1)x_n z^{n-2},$$

so the gradient constraints (6.43) are linear equality and inequality constraints on the variable $x \in \mathbf{R}^n$. The polynomial function fitting result from solving the problem (6.41) with the interpolation constraints (6.42) and the gradient constraints (6.43) is shown as the dashdotted curve in figure 6.4.

Finally, we consider the problem of finding a polynomial function of the form (6.40) that bounds all the data points from below. This bounding condition can be expressed as the linear inequality constraints

$$f(z_i) \leq y_i, \quad i = 1, \ldots, m. \tag{6.44}$$

Solving the problem (6.41) with the bounding constraints (6.44) gives us the thinner solid curve shown in the bottom of figure 6.4.

6.5 Relaxations

6.5.1 Definition and basic properties

Relaxation is an operation on optimization problems (or usually, on the constraints) that transforms a given problem into another one that is easier to solve, while preserving some properties of the original problem. For instance, let

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & x \in \mathcal{P} \end{array} \tag{6.45}$$

be an optimization problem with variable $x \in \mathbf{R}^n$, objective function $f_0 \colon \mathbf{R}^n \to \mathbf{R}$, and constraint set $\mathcal{P} \subseteq \mathbf{R}^n$. Let $\mathcal{Q} \subseteq \mathbf{R}^n$ be another constraint set such that $\mathcal{P} \subseteq \mathcal{Q}$, then the optimization problem

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & x \in \mathcal{Q} \end{array} \tag{6.46}$$

is called a *relaxation* of the original problem (6.45). The relaxed constraint set $\mathcal{Q}$ is usually chosen to be a convex set, so that the relaxed problem (6.46) is a convex optimization problem (provided the objective function f_0 is convex).

Roughly speaking, when introducing a relaxation to an optimization problem, we are *ignoring* some of the constraints in the problem. As an extreme example, ignoring all the constraints in the original problem (6.45) is equivalent to taking $\mathcal{Q} = \mathbf{R}^n$, and thus the resulting relaxed problem (6.46) is an unconstrained optimization problem.

Lower bounding property

One of the most important properties of relaxations is that an optimal value of a relaxation provides a *lower bound* on the optimal value of the original problem. In particular, let $p^\star$ and $q^\star$ be optimal values of the original problem (6.45) and its relaxation (6.46), respectively, then we have

$$q^\star = \inf_{x \in \mathcal{Q}} f_0(x) \leq \inf_{x \in \mathcal{P}} f_0(x) = p^\star, \tag{6.47}$$

where the inequality follows from the fact that $\mathcal{P} \subseteq \mathcal{Q}$.

The lower bounding property of relaxations given by (6.47) is particularly useful when the original problem (6.45) is a nonconvex optimization problem, but the relaxation (6.46) is convex. In this case, we can solve the relaxation (6.46) efficiently to obtain a lower bound on the optimal value of the original problem (6.45). This lower bound can then be used to evaluate the quality of any feasible point of the original problem. Specifically, if a feasible point $x \in \mathcal{P}$ of the problem (6.45) leads to an objective value $f_0(x)$ such that the gap $|f_0(x) - q^\star|$ is small, then the point x is likely to be close to an optimal point of the original problem (6.45).

Moreover, solving the relaxation (6.46) can sometimes directly give us an optimal point of the original problem (6.45). Specifically, let $x^\star \in \mathcal{Q}$ be an optimal point of the relaxed problem (6.46), *i.e.*,

$$q^\star = \inf_{x \in \mathcal{Q}} f_0(x) = f_0(x^\star).$$

If $x^\star$ happens to be a feasible point of the original problem (6.45), *i.e.*, $x^\star \in \mathcal{P}$, then $x^\star$ must be an optimal point of the original problem (6.45), *i.e.*,

$$p^\star = \inf_{x \in \mathcal{P}} f_0(x) = f_0(x^\star) = q^\star.$$

In other words, the inequality (6.47) holds with equality. To show this, noticing that if $x^\star \in \mathcal{P}$, then we have

$$p^\star = \inf_{x \in \mathcal{P}} f_0(x) \leq f_0(x^\star) = q^\star.$$

However, by the lower bounding property of relaxations given by the inequality (6.47), we also have

$$q^\star \leq p^\star,$$

so it follows that $p^\star = q^\star = f_0(x^\star)$, and thus $x^\star$ is an optimal point of the original problem (6.45).

Remark 6.4 *Relaxations on the objective.* The idea of relaxations can also be applied to the objective function of an optimization problem of the form (6.45). For example, let $g_0 : \mathbf{R}^n \to \mathbf{R}$ be a function such that $g_0(x) \leq f_0(x)$ for all $x \in \mathcal{P}$, then the problem

$$\begin{array}{ll} \text{minimize} & g_0(x) \\ \text{subject to} & x \in \mathcal{P} \end{array} \tag{6.48}$$

can be considered as a special type of relaxation of the problem (6.45), and its optimal value also provides a lower bound on the optimal value of the original problem. The function g_0 in (6.48) is usually chosen to be a convex function, *e.g.*, the convex envelope of f_0 on the constraint set $\mathcal{P}$, so that the resulting relaxed problem is a convex optimization problem (provided the constraints are convex).

For this type of relaxations where the objective function has been changed, in the most general case, we cannot say much about the relationship between the optimal points (and values, besides the lower bounding property) of the original problem and its relaxation, since the lower bound on the optimal value of the original problem (6.45) obtained from the relaxation (6.48) might be very loose, and the optimal points of the relaxed problem might be very different from those of the original problem. However, there are some special cases where the optimal values and points of the original problem and its relaxation are closely related. One famous example is called the *Lagrangian relaxation*, which we will see soon in §6.6.

In practice, the problem (6.48) is sometimes referred to as a *surrogate problem* of the original problem (6.45), and the function g_0 is called a *surrogate objective function* of the original objective function f_0.

Sensitivity analysis

Another important property of relaxations is that the optimal value of a relaxation can be used to analyze the sensitivity of the original problem to perturbations on the constraints. As an example, consider an inequality constrained optimization problem of the form

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \ldots, m \end{array} \tag{6.49}$$

with variable $x \in \mathbf{R}^n$, and let

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq u_i, \quad i = 1, \ldots, m \end{array} \tag{6.50}$$

be a perturbed version of the original problem, where $u \in \mathbf{R}^m$ is the perturbation parameter. When $u = 0$, the perturbed problem (6.50) reduces to the original problem (6.49). If $u_i > 0$, then the ith inequality constraint in (6.49) is *relaxed*, and if $u_i < 0$, then the ith inequality constraint is *tightened*.

Let $p^\star$ be the optimal value of the original problem (6.49), and define the function $q^\star : \mathbf{R}^m \to \mathbf{R}$ as

$$q^\star(u) = \inf \left\{ f_0(x) \;\middle|\; \begin{array}{l} x \in \bigcap_{i=0}^m \mathbf{dom}\, f_i \\ f_i(x) \leq u_i, \quad i = 1, \ldots, m \end{array} \right\},$$

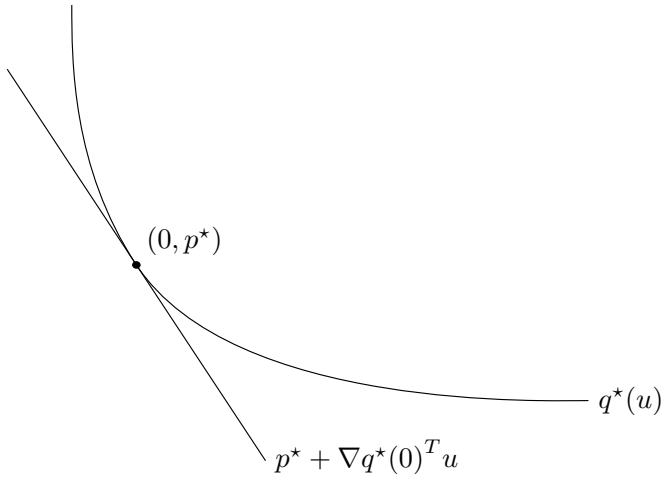

Figure 6.5 Optimal value $q^\star(u)$ of the convex perturbed problem (6.50) as a function of the perturbation parameter $u \in \mathbf{R}^m$. For $u = 0$, we have $q^\star(0) = p^\star$, where $p^\star$ is the optimal value of the original convex problem (6.49). The gradient $\nabla q^\star(0)$ gives the local sensitivity of the original problem to perturbations on the constraint.

which gives the optimal value of the perturbed problem (6.50), parameterized by $u \in \mathbf{R}^m$. Note that the function $q^\star$ can take the value ∞ for some $u \in \mathbf{R}^m$, which means that the perturbed problem (6.50) is infeasible for these values of u. If the original problem (6.49) is convex, then it can be shown that the function $q^\star$ is a convex function of u; see exercise 6.3.

Noticing that $q^\star(0) = p^\star$, we can analyze the sensitivity of the original problem (6.49) to perturbations on the constraints by studying the behavior of the function $q^\star$ in a neighborhood of $u = 0$. For example, suppose the ith constraint in (6.49) is relaxed by a small amount $u_i > 0$, then if the optimal value $q^\star(u)$ decreases significantly compared to $p^\star$, then the original problem (6.49) is very sensitive to relaxations on the ith constraint. Similar analysis also applies to the case where the ith constraint is tightened by a small amount $u_i < 0$. In the limit, if the function $q^\star$ is differentiable at $u = 0$, then the gradient $\nabla q^\star(0)$ gives the local sensitivity of the problem (6.49) to perturbations on the constraints. These ideas are illustrated in figure 6.5.

We will see later in §6.6 that the perturbation (6.50) is the essence of *Lagrangian relaxation*, which actually provides an approach to compute the gradient $\nabla q^\star(0)$, so that it is practically actionable to analyze the local sensitivity of the original problem (6.49) to perturbations on the constraints.

6.5.2 Examples

In this section, we describe several typical examples of relaxations that appear in different contexts and applications.

Cardinality and rank constraints

Consider the cardinality constrained optimization problem

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & \mathbf{card}\, x \leq k, \quad \|x\|_\infty \leq 1, \end{array} \tag{6.51}$$

where $x \in \mathbf{R}^n$ is the variable. Here, the cardinality constraint $\mathbf{card}\, x \leq k$ requires the vector x to have at most k nonzero entries, and the bounding constraint $\|x\|_\infty \leq 1$ requires the absolute value of each entry of x to be at most 1. Since the cardinality function is not convex, the problem (6.51) is nonconvex and generally hard to solve. However, under this ℓ_∞-norm ball constraint, the convex ℓ_1-norm function is the convex envelope of the cardinality function (see example 2.11 on page 38), *i.e.*, we have

$$\|x\|_1 \leq \mathbf{card}\, x$$

for all $x \in \mathbf{R}^n$ such that $\|x\|_\infty \leq 1$. Therefore, a convex relaxation of the cardinality constrained problem (6.51) is given by

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & \|x\|_1 \leq k, \quad \|x\|_\infty \leq 1, \end{array} \tag{6.52}$$

which is a convex optimization problem if the objective function f_0 is convex. To see that the feasible set of the original problem (6.51) is contained in the feasible set of the relaxed problem (6.52), let $x \in \mathbf{R}^n$ be a vector such that $\mathbf{card}\, x \leq k$ and $\|x\|_\infty \leq 1$, then we have

$$\|x\|_1 = \sum_{i=1}^n |x_i| \leq \mathbf{1}^T p = \mathbf{card}\, x \leq k, \qquad p = \begin{cases} 1, & x_i \neq 0 \\ 0, & \text{otherwise}, \end{cases}$$

so the vector x is also a feasible point of the relaxed problem (6.52). Obviously, the converse does not hold, so the feasible set of (6.52) is a strict superset of the feasible set of (6.51).

Similar ideas extend to optimization problems with matrix variable. Consider the rank constrained optimization problem

$$\begin{array}{ll} \text{minimize} & f_0(X) \\ \text{subject to} & \mathbf{rank}\, X \leq k, \quad \|X\|_2 \leq 1, \end{array}$$

where $X \in \mathbf{R}^{m \times n}$ is the variable, and $\|\cdot\|_2$ is the *spectral norm*, *i.e.*, the largest singular value of a matrix. A convex relaxation of this problem is given by

$$\begin{array}{ll} \text{minimize} & f_0(X) \\ \text{subject to} & \|X\|_* \leq k, \quad \|X\|_2 \leq 1, \end{array}$$

where $\|\cdot\|_*$ is the *nuclear norm*, *i.e.*, the sum of singular values of a matrix, which is the convex envelope of the rank function under the unit spectral norm ball

constraint. We could also verify that the feasible set of the original rank constrained problem is contained in the feasible set of its nuclear norm relaxation. Let $X \in \mathbf{R}^{m\times n}$ be a matrix such that $\mathbf{rank}\, X \leq k$ and $\|X\|_2 \leq 1$, then we have

$$\|X\|_* = \sum_{i=1}^{k} \sigma_i(X) \leq k\sigma_1(X) = k\|X\|_2 \leq k,$$

where $\sigma_i(X)$ is the ith largest singular value of X for $i = 1, \ldots, \min\{m, n\}$.

Boolean linear programs

Consider the optimization problem

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax \preceq b \\ & x_i \in \{0, 1\}, \quad i = 1, \ldots, n, \end{array} \tag{6.53}$$

where $x \in \mathbf{R}^n$ is the variable. This problem is sometimes called a *boolean linear program*, since each entry of the variable x is required to be either 0 or 1. When n is not too large, the boolean linear program (6.53) can be solved exactly by searching over all the 2^n possible values of x. However, when n is large, the problem (6.53) is generally very hard to solve.

A common relaxation of the boolean linear program (6.53) is given by

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax \preceq b \\ & 0 \preceq x \preceq 1, \end{array} \tag{6.54}$$

where the boolean constraints $x_i \in \{0, 1\}$ are relaxed to the box constraints $0 \leq x_i \leq 1$ for $i = 1, \ldots, n$. This relaxation is sometimes called the *LP relaxation* of the boolean linear program (6.53). The LP relaxation (6.54), as the name suggests, is a linear program, and hence can be solved much faster than the original problem (6.53).

Noticing that the problem (6.54) yields a lower bound on the optimal value of the original problem (6.53), as a result, if the relaxation (6.54) is infeasible, then the original problem (6.53) must also be infeasible. Moreover, if the optimal point of the relaxed problem (6.54) happens to be a boolean vector, *i.e.*, satisfies $x_i \in \{0, 1\}$ for all $i = 1, \ldots, n$, then it is also an optimal point of the original problem (6.53).

Similar ideas extend to other problems with discrete constraints, *e.g.*, the *integer program* described in example 2.23.

Two-way partitioning

Consider the (possibly nonconvex) optimization problem

$$\begin{array}{ll} \text{minimize} & x^T W x \\ \text{subject to} & x_i \in \{-1, 1\}, \quad i = 1, \ldots, n, \end{array} \tag{6.55}$$

where $x \in \mathbf{R}^n$ is the variable, and $W \in \mathbf{S}^n$ is a given matrix. This problem is sometimes called a *two-way partitioning problem*, since each feasible $x \in \mathbf{R}^n$ corresponds to a partition of the set $\{1, \ldots, n\}$ into two subsets, given by

$$\{1, \ldots, n\} = \{i \mid x_i = 1\} \cup \{i \mid x_i = -1\}.$$

In this context, the matrix W can be interpreted as a weight matrix that encodes the pairwise relationships between the elements in the set $\{1, \ldots, n\}$. In particular, the value W_{ij} is the cost of putting the elements i and j in the same subset of the partition, while $-W_{ij}$ is the cost of putting them in different subsets. Hence, the objective function $x^T W x$ gives the total cost of the partition corresponding to x, and the problem (6.55) is to find a partition that minimizes this cost. Similar to the boolean linear program (6.53), the two-way partitioning problem (6.55) has a finite feasible set and is generally hard to solve when n is large.

To obtain a convex relaxation of the problem (6.55), we first notice that the integer constraints in (6.55) can be equivalently expressed as $x_i^2 = 1$ for all $i = 1, \ldots, n$. Then we introduce a new variable $X = xx^T \in \mathbf{S}_+^n$, which is a rank one positive semidefinite matrix, so the problem (6.55) is equivalent to

$$\begin{array}{ll} \text{minimize} & \mathbf{tr}(WX) \\ \text{subject to} & X \succeq 0, \quad \mathbf{rank}\, X = 1 \\ & X_{ii} = 1, \quad i = 1, \ldots, n, \end{array} \tag{6.56}$$

where $X \in \mathbf{S}^n$ is the variable. By the formulation (6.56), it is easily seen that the nonconvexity of the two-way partitioning problem is due to the rank one constraint on the variable X. Therefore, a simple convex relaxation of the problem (6.56) can be obtained by ignoring the rank constraint, which is given by

$$\begin{array}{ll} \text{minimize} & \mathbf{tr}(WX) \\ \text{subject to} & X \succeq 0 \\ & X_{ii} = 1, \quad i = 1, \ldots, n. \end{array} \tag{6.57}$$

Since the relaxed problem (6.57) is a semidefinite program, this type of relaxation is sometimes called a *semidefinite relaxation*. Again, if an optimal point of the relaxed problem (6.57) happens to be a rank one matrix, then it must also be optimal for (6.56) (and hence be optimal for the original problem (6.55)).

6.6 Lagrangian relaxation and duality

This section describes *Lagrangian relaxation* and *duality* in optimization problems. Lagrangian relaxation is a special type of relaxations and has lots of very useful theoretical and practical properties. *Analysis* related to Lagrangian relaxation and duality is a fundamental topic in optimization theory, which we will not pursue in any depth. We refer the readers interested in these topics to the references listed at the end of this chapter.

6.6.1 The Lagrangian

We consider an optimization problem of the form

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \ldots, m \\ & h_i(x) = 0, \quad i = 1, \ldots, p, \end{array} \tag{6.58}$$

where $x \in \mathbf{R}^n$ is the variable, $f_0, f_1, \ldots, f_m \colon \mathbf{R}^n \to \mathbf{R}$ are the objective and inequality constraint functions, and $h_1, \ldots, h_p \colon \mathbf{R}^n \to \mathbf{R}$ are the equality constraint functions. Note that we do not assume any convexity properties of the functions $f_0, f_1, \ldots, f_m$ and $h_1, \ldots, h_p$ in the problem (6.58), so the problem (6.58) is a general nonlinear optimization problem.

The *Lagrangian* of the problem (6.58) is defined as the function $L \colon \mathbf{R}^n \times \mathbf{R}^m \times \mathbf{R}^p \to \mathbf{R}$ given by

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x).$$

Let $\mathcal{D} = \bigcap_{i=0}^{m} \mathbf{dom}\, f_i \cap \bigcap_{i=1}^{p} \mathbf{dom}\, h_i$, then the domain of L is given by

$$\mathbf{dom}\, L = \mathcal{D} \times \mathbf{R}^m \times \mathbf{R}^p.$$

For each $i = 1, \ldots, m$, the number λ_i is called the *Lagrange multiplier* associated with the ith inequality constraint $f_i(x) \leq 0$, and for each $i = 1, \ldots, p$, the number ν_i is called the Lagrange multiplier associated with the ith equality constraint $h_i(x) = 0$. The vectors $\lambda \in \mathbf{R}^m$ and $\nu \in \mathbf{R}^p$ are called the *dual variables* or *Lagrange multiplier vectors*, associated with the problem (6.58).

6.6.2 The Lagrange dual function

The *Lagrange dual function* (or just *dual function*) $g \colon \mathbf{R}^m \times \mathbf{R}^p \to \mathbf{R}$ associated with the problem (6.58) is defined as the partial infimum of the Lagrangian L over the variable $x \in \mathcal{D}$, *i.e.*,

$$g(\lambda, \nu) = \inf_{x \in \mathcal{D}} L(x, \lambda, \nu) = \inf_{x \in \mathcal{D}} \left(f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x) \right).$$

If for some values of the dual variables $\lambda \in \mathbf{R}^m$ and $\nu \in \mathbf{R}^p$, the function $L(x, \lambda, \nu)$ is unbounded below over $x \in \mathcal{D}$, then we have $g(\lambda, \nu) = -\infty$.

One of the most important properties of the dual function g is that it is always a *concave* function of the dual variables λ and ν, even if the original problem (6.58) is nonconvex. To see this, notice that for each fixed $x \in \mathcal{D}$, the function $L(x, \lambda, \nu)$ is an affine function of λ and ν, and thus the dual function g is the pointwise infimum of a family of affine functions, which is concave.

Another important property of the dual function g is that it can provide a *lower bound* on the optimal value of the original problem (6.58). Specifically, let $p^\star$ be an optimal value of the original problem (6.58), then for any $\lambda \succeq 0$ and ν, we have

$$g(\lambda, \nu) \leq p^\star. \tag{6.59}$$

When the dual function is unbounded below, *i.e.*, $g(\lambda, \nu) = -\infty$ for some $\lambda \succeq 0$ and ν, the inequality (6.59) holds trivially, so it only gives us useful information about $p^\star$ when $\lambda \succeq 0$ and $g(\lambda, \nu) > -\infty$ (*i.e.*, $(\lambda, \nu) \in \mathbf{dom}\, g$). We refer to the dual variables λ and ν that satisfy $\lambda \succeq 0$ and $(\lambda, \nu) \in \mathbf{dom}\, g$ as *dual feasible* (for reasons that will become clear later).

Remark 6.5 *Proof of lower bounds on the optimal value.* To show the inequality (6.59), let $\tilde{x} \in \mathcal{D}$ be a feasible point of the original problem (6.58), *i.e.*,

$$f_i(\tilde{x}) \leq 0, \quad i = 1, \ldots, m, \qquad h_i(\tilde{x}) = 0, \quad i = 1, \ldots, p,$$

then for all $\lambda \in \mathbf{R}^m_+$ and $\nu \in \mathbf{R}^p$, we have

$$\sum_{i=1}^m \lambda_i f_i(\tilde{x}) \leq 0 \quad \text{and} \quad \sum_{i=1}^p \nu_i h_i(\tilde{x}) = 0,$$

which implies that

$$\sum_{i=1}^m \lambda_i f_i(\tilde{x}) + \sum_{i=1}^p \nu_i h_i(\tilde{x}) \leq 0.$$

Therefore, we have the following inequality for the Lagrangian L:

$$L(\tilde{x}, \lambda, \nu) = f_0(\tilde{x}) + \sum_{i=1}^m \lambda_i f_i(\tilde{x}) + \sum_{i=1}^p \nu_i h_i(\tilde{x}) \leq f_0(\tilde{x}). \tag{6.60}$$

It then follows that

$$g(\lambda, \nu) = \inf_{x \in \mathcal{D}} L(x, \lambda, \nu) \leq L(\tilde{x}, \lambda, \nu) \leq f_0(\tilde{x}).$$

Since $g(\lambda, \nu) \leq f_0(\tilde{x})$ for all feasible points $\tilde{x} \in \mathcal{D}$, we have

$$g(\lambda, \nu) \leq \inf_{x \in \mathcal{D}} \left\{ f_0(x) \;\middle|\; \begin{array}{ll} f_i(x) \leq 0, & i = 1, \ldots, m \\ h_i(x) = 0, & i = 1, \ldots, p \end{array} \right\} = p^\star,$$

which completes the proof.

Example 6.12 *Lagrangian and the dual function.* Consider a simple nonlinear optimization problem

$$\begin{array}{ll} \text{minimize} & f_0(x) = x^4 - x^2 - (1/5)x + 1 \\ \text{subject to} & f_1(x) = (1/4)\log(1 + 20x^2) - 0.5 \leq 0 \end{array} \tag{6.61}$$

with variable $x \in \mathbf{R}$. The objective and constaint functions of this problem are shown as the solid and dashdotted curves in figure 6.6, respectively. The interval between the two vertical dotted lines in the figure is the feasible set of the problem. The optimal point of the problem (6.61) is obviously the right endpoint of the feasible interval, which is shown as the solid dot.

The Lagrangian of the problem (6.61) is given by

$$\begin{array}{rcl} L(x, \lambda) & = & f_0(x) + \lambda f_1(x) \\ & = & x^4 - x^2 - (1/5)x + 1 + \lambda((1/4)\log(1 + 20x^2) - 0.5), \end{array}$$

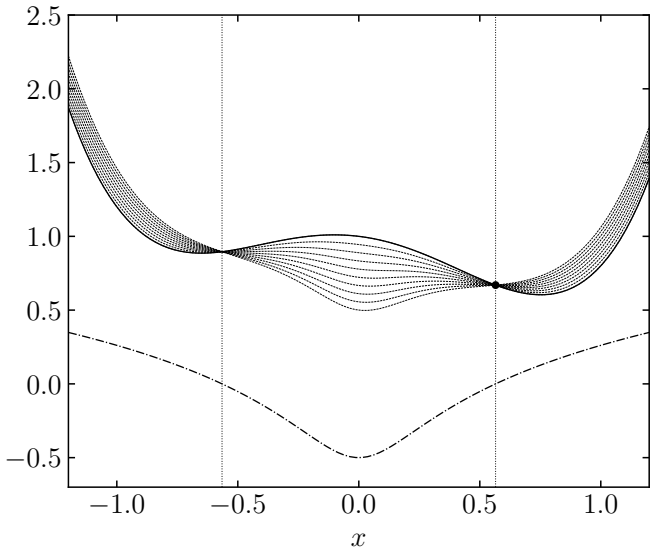

Figure 6.6 Objective function (solid curve) and constraint function (dashdotted curve) of the problem (6.61). The feasible set corresponds to the interval between the two vertical dotted lines, and the optimal point is shown as the solid dot. The dashed curves show the Lagrangian $L(x, \lambda)$ of the problem (6.61) for different values of $\lambda = 0.1, 0.2, \ldots, 1$.

where $\lambda \in \mathbf{R}$ is the dual variable. By (6.60), for all feasible point x and $\lambda \geq 0$, we have $L(x, \lambda) \leq f_0(x)$, *i.e.*, for all dual feasible points λ, the Lagrangian is always an underestimator of the objective function within the feasible set. This is illustrated in figure 6.6, where the dashed curves depict the Lagrangian $L(x, \lambda)$ for different values of $\lambda = 0.1, 0.2, \ldots, 1$.

Figure 6.7 shows the dual function $g(\lambda) = \inf_{x \in \mathbf{R}} L(x, \lambda)$ of the problem (6.61), where the horizontal dashed line corresponds to the optimal value of the original problem. Firstly, we may notice that even if the objective f_0 and the constraint function f_1 are both nonconvex, the dual function g is a concave function of λ. Moreover, it is clear from the figure that for all $\lambda \geq 0$, the dual function g always provides a lower bound on the optimal value of the original problem.

Relaxation interpretation

The Lagrange dual function $g\colon \mathbf{R}^m \times \mathbf{R}^p \to \mathbf{R}$ of the problem (6.58) corresponds to the optimal value of the following problem:

$$\text{minimize} \quad f_0(x) + \sum_{i=1}^m \lambda_i f_i(x) + \sum_{i=1}^p \nu_i h_i(x) \tag{6.62}$$

with variable $x \in \mathbf{R}^n$ and hyperparameters $\lambda \in \mathbf{R}^m$ and $\nu \in \mathbf{R}^p$, which can be considered as a special type of relaxation of the original problem (6.58). To explain this, we introduce two auxiliary variables $u \in \mathbf{R}^m$ and $v \in \mathbf{R}^p$, so the problem

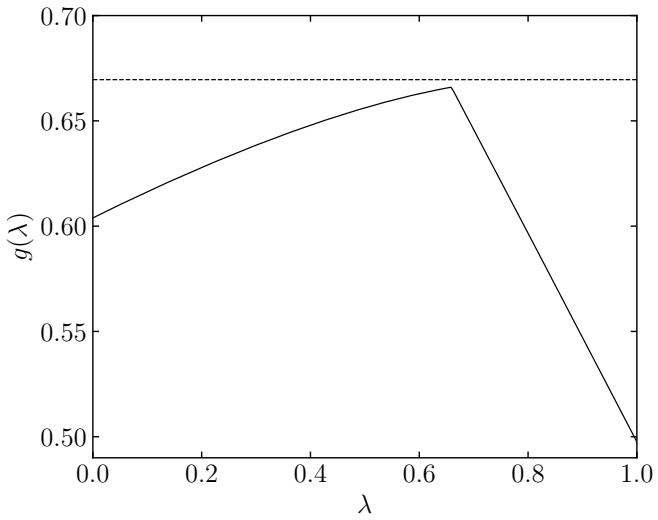

Figure 6.7 The dual function g (solid curve) of the problem (6.61). The dashed line corresponds to the optimal value of the original problem.

(6.62) can be reformulated as

$$\begin{array}{ll} \text{minimize} & f_0(x) + \lambda^T u + \nu^T v \\ \text{subject to} & f_i(x) \leq u_i, \quad i = 1, \ldots, m \\ & h_i(x) = v_i, \quad i = 1, \ldots, p, \end{array} \tag{6.63}$$

where, note that, the variables are $x \in \mathbf{R}^n$, $u \in \mathbf{R}^m$, and $v \in \mathbf{R}^p$. Obviously, this problem is always feasible, since for any $x \in \mathbf{R}^n$, there always exist some u and v to satisfy the constraints. According to this formulation, we can see that rather than hardly enforces the original constraints $f_i(x) \leq 0$ and $h_i(x) = 0$ in the problem (6.58), the problem (6.63) allows some violation of these constraints, which is measured by the variables u and v. The total constraint violation, or relaxation, is then penalized in the objective by the linear penalty $\lambda^T u + \nu^T v$, with the penalty strength controlled by the dual variables λ and ν.

Linear approximation interpretation

The problem (6.62) can also be interpreted as a linear approximation of the original problem (6.58). To see this, we first reformulate the original problem (6.58) as

$$\text{minimize} \quad f_0(x) + \textstyle\sum_{i=1}^m I_-(f_i(x)) + \sum_{i=1}^p I_0(h_i(x)) \tag{6.64}$$

with variable $x \in \mathbf{R}^n$, where $I_-, I_0 \colon \mathbf{R} \to \mathbf{R}$ are the indicator functions of the sets $\mathbf{R}_-$ and $\{0\}$, respectively, *i.e.*,

$$I_-(t) = \begin{cases} 0, & t \leq 0 \\ \infty, & \text{otherwise} \end{cases} \qquad \text{and} \qquad I_0(t) = \begin{cases} 0, & t = 0 \\ \infty, & \text{otherwise.} \end{cases}$$

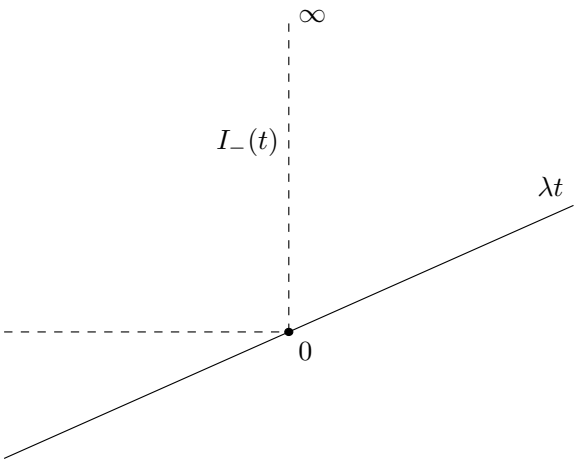

Figure 6.8 Graph of the indicator function I_- (shown dashed) of $\mathbf{R}_-$ and its linear approximation λt (shown solid) for some $\lambda > 0$.

Then evaluating the objective of the problem (6.64) at a feasible point x of the original problem (6.58) yields exactly the same value as $f_0(x)$, and ∞ otherwise. In other words, the indicator functions I_- and I_0 simply add a 'barrier' to $f_0(x)$ that prevents any violation of the constraints, so that the constraints are now implicit in the objective function. Therefore, we can see that the problem (6.62) is obtained by replacing the indicator functions $I_-(f_i(x))$ and $I_0(h_i(x))$ in (6.64) with their linear approximations $\lambda_i f_i(x)$ and $\nu_i h_i(x)$, respectively. Since for all $\lambda \in \mathbf{R}^m_+$ and $\nu \in \mathbf{R}^p$, we have $\lambda_i t \leq I_-(t)$ and $\nu_i t \leq I_0(t)$ for all $t \in \mathbf{R}$, the problem (6.62) yields a lower bound on the optimal value of the original problem (6.64). This interpretation is illustrated in figure 6.8.

From the above interpretations, we may see that the relaxation (6.62) from the dual function of the problem (6.58) is rather poor. Nevertheless, it at least provides a lower bound on the optimal value of the original problem, and we will see in the next few sections that this lower bound can actually be tight in many useful cases.

6.6.3 The Lagrange dual problem

Since the dual function $g(\lambda, \nu)$ for all $\lambda \succeq 0$ and ν provides a lower bound on the optimal value of the original problem (6.58), it is natural to ask what is the *best*, or *greatest* lower bound that we can obtain from the dual function. This leads to the following optimization problem, which is called the *Lagrange dual problem* associated with the problem (6.58):

$$\begin{array}{ll} \text{maximize} & g(\lambda, \nu) \\ \text{subject to} & \lambda \succeq 0, \end{array} \tag{6.65}$$

where $\lambda \in \mathbf{R}^m$ and $\nu \in \mathbf{R}^p$ are the variables. In this context, the original problem (6.58) is called the *primal problem*. The feasible set of the dual problem (6.65) is

given by

$$\{(\lambda, \nu) \in \mathbf{dom}\, g \mid \lambda \succeq 0\},$$

so the definition of *dual feasible* points on page 226 now makes more sense. An optimal value $d^\star$ of the dual problem (6.65) is called the *dual optimal value*, and the corresponding optimal point $(\lambda^\star, \nu^\star)$ is called the *dual optimal point*, or *optimal Lagrange multipliers*.

Note that since the dual function g is always a concave function in the dual variables λ and ν, the dual problem (6.65) is a convex optimization problem, even if the primal problem (6.58) is nonconvex.

Weak duality

A direct consequence of the lower bounding property (6.59) of the dual function g on the dual problem (6.65) is that the dual optimal value $d^\star$ must be a lower bound on the primal optimal value $p^\star$, *i.e.*,

$$d^\star \leq p^\star, \tag{6.66}$$

which always holds regardless of the convexity of the primal problem (6.58). This is called the *weak duality* property of the primal and dual problems. In this case, the gap between the primal and dual optimal values, *i.e.*,

$$p^\star - d^\star,$$

is called the *optimal duality gap*, which is always nonnegative and represents the greatest lower bound on the optimal value of the original problem (6.58) that can be obtained from the Lagrange dual function. This method of finding a lower bound on the optimal value is called *Lagrangian relaxation*.

Weak duality has many useful applications in practice. For example, it provides an actionable approach to obtain a lower bound on the optimal value of some nonconvex problem by solving its dual.

Example 6.13 *Two-way partitioning.* There are several options to obtain a lower bound on the optimal value of the nonconvex two-way partitioning problem

$$\begin{array}{ll} \text{minimize} & x^T W x \\ \text{subject to} & x_i^2 = 1, \quad i = 1, \ldots, n, \end{array} \tag{6.67}$$

where $x \in \mathbf{R}^n$ is the variable and $W \in \mathbf{S}^n$ is a given matrix. Page 223 describes one option via semidefinite relaxation. Here, we introduce another way for lower bounding the problem (6.67) via Lagrangian relaxation.

The Lagrangian of the problem (6.67) is given by

$$L(x, \nu) = x^T W x + \sum_{i=1}^n \nu_i (x_i^2 - 1) = x^T (W + \mathbf{diag}(\nu)) x - \mathbf{1}^T \nu,$$

so the dual function is expressed as

$$g(\nu) = \inf_x L(x, \nu) = \inf_x x^T (W + \mathbf{diag}(\nu)) x - \mathbf{1}^T \nu.$$

Noticing that the function $x^T(W + \mathbf{diag}(\nu))x$ is a quadratic form, we have

$$g(\nu) = \begin{cases} -\mathbf{1}^T\nu, & W + \mathbf{diag}(\nu) \succeq 0 \\ -\infty, & \text{otherwise.} \end{cases}$$

Therefore, the dual problem of the problem (6.67) is given by

$$\begin{array}{ll} \text{maximize} & -\mathbf{1}^T\nu \\ \text{subject to} & W + \mathbf{diag}(\nu) \succeq 0, \end{array}$$

where $\nu \in \mathbf{R}^n$ is the variable. This dual problem is a semidefinite program, and hence can be solved efficiently, which yields a lower bound on the optimal value of the original problem (6.67).

The weak duality property is also related to the feasibility of the primal and dual problems. For example, if the primal problem (6.58) is unbounded below, *i.e.*, $p^\star = -\infty$, then we must have $d^\star = -\infty$, *i.e.*, the dual problem is infeasible. Conversely, if the dual problem (6.65) is unbounded above, *i.e.*, $d^\star = \infty$, then we must have $p^\star = \infty$, *i.e.*, the primal problem (6.58) must be infeasible.

Strong duality

If for some problem (6.58), the inequality (6.66) holds with equality, *i.e.*,

$$d^\star = p^\star,$$

or in other words, the optimal duality gap is zero, then we say that *strong duality* holds between the primal and dual problems. This says that, the greatest lower bound on the optimal value of the original problem (6.58) that can be obtained from the dual function is actually tight, and thus we can obtain the optimal value of the original problem by solving its dual problem.

We should note that strong duality does not always hold for all problems in the form (6.58). The conditions that guarantee strong duality are called *constraint qualifications*, which we will not discuss in any depth here. As a simple case, if the primal problem (6.58) is a linear program, then strong duality is equivalent to feasibility of the primal problem. Besides this, we could (very) roughly say that strong duality *usually* holds when the primal problem is convex; see also the references.

One useful consequence of strong duality is that, in this case, the dual optimal value $d^\star$ provides a 'certificate' of optimality for the primal optimal value $p^\star$. Specifically, when strong duality holds for the problem (6.58), suppose some feasible point $x^\star$ of the primal problem (6.58) leads to the objective value $f_0(x^\star)$ that is equal to the dual optimal value $d^\star$, then we can conclude that $x^\star$ must be an optimal point of the primal problem, and the corresponding optimal value is $p^\star = f_0(x^\star)$.

In the most general case (where we do not make any assumptions about strong duality), suppose the point $\tilde{x}$ is primal feasible and the point $(\tilde{\lambda}, \tilde{\nu})$ is dual feasible, then we have

$$p^\star \in [g(\tilde{\lambda}, \tilde{\nu}), f_0(\tilde{x})], \qquad d^\star \in [g(\tilde{\lambda}, \tilde{\nu}), f_0(\tilde{x})], \tag{6.68}$$

where the length of the interval $[g(\tilde{\lambda},\tilde{\nu}), f_0(\tilde{x})]$, given by

$$f_0(\tilde{x}) - g(\tilde{\lambda},\tilde{\nu}),$$

is called the *duality gap* between the primal and dual feasible points $\tilde{x}$ and $(\tilde{\lambda},\tilde{\nu})$. The results in (6.68) implies that if the primal and dual feasible points $\tilde{x}$ and $(\tilde{\lambda},\tilde{\nu})$ lead to zero duality gap, *i.e.*, satisfy $g(\tilde{\lambda},\tilde{\nu}) = f_0(\tilde{x})$, then we must have $p^\star = d^\star = g(\tilde{\lambda},\tilde{\nu}) = f_0(\tilde{x})$, and thus $\tilde{x}$ is an optimal point of the primal problem, and $(\tilde{\lambda},\tilde{\nu})$ is an optimal point of the dual problem.

Strong duality also has useful implications regarding *sensitivity analysis* of the optimal value of the primal problem with respect to perturbations of the constraints. The following example illustrates these ideas.

Example 6.14 *Sensitivity analysis.* Consider a perturbed version of the original problem (6.58):

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \le u_i, \quad i = 1,\dots,m \\ & h_i(x) = v_i, \quad i = 1,\dots,p, \end{array} \tag{6.69}$$

where $x \in \mathbf{R}^n$ is the variable. Let $q^\star : \mathbf{R}^m \times \mathbf{R}^p \to \mathbf{R}$ be the function that gives the optimal value of the perturbed problem (6.69), and let $\mathcal{D} = \bigcap_{i=0}^m \mathbf{dom}\, f_i \cap \bigcap_{i=1}^p \mathbf{dom}\, h_i$, then we have

$$q^\star(u,v) = \inf \left\{ f_0(x) \;\middle|\; \begin{array}{l} x \in \mathcal{D} \\ f_i(x) \le u_i,\ i = 1,\dots,m \\ h_i(x) = v_i,\ i = 1,\dots,p \end{array} \right\}, \tag{6.70}$$

and $q^\star(0,0) = p^\star$ is the optimal value of the original problem (6.58).

Let $(\lambda^\star,\nu^\star)$ be an optimal point of the dual (6.65) of the original problem (6.58). Assume that strong duality holds, and the dual optimal value is $d^\star$ is finite, then we have

$$q^\star(0,0) = p^\star = d^\star = g(\lambda^\star,\nu^\star) = \inf_{x\in\mathcal{D}} L(x,\lambda^\star,\nu^\star),$$

which implies that

$$q^\star(0,0) \le f_0(x) + \sum_{i=1}^m \lambda_i^\star f_i(x) + \sum_{i=1}^p \nu_i^\star h_i(x)$$

for all $x \in \mathcal{D}$. Now suppose $x \in \mathcal{D}$ is a feasible point of the perturbed problem (6.69), *i.e.*,

$$f_i(x) \le u_i, \quad i = 1,\dots,m, \qquad h_i(x) = v_i, \quad i = 1,\dots,p,$$

then by the above inequality (and notice that $\lambda^\star \succeq 0$), we have

$$q^\star(0,0) \le f_0(x) + \lambda^{\star T} u + \nu^{\star T} v,$$

i.e.,

$$f_0(x) \ge q^\star(0,0) - \lambda^{\star T} u - \nu^{\star T} v.$$

Hence, according to the definition (6.70) of the function $q^\star$, we conclude that

$$q^\star(u,v) \ge q^\star(0,0) - \lambda^{\star T} u - \nu^{\star T} v. \tag{6.71}$$

Noticing that we did not make any assumptions regarding the perturbation parameters, this inequality holds *globally* for all $u \in \mathbf{R}^m$ and $v \in \mathbf{R}^p$.

We may obtain many useful information from the inequality (6.71). First of all, it says that the function $q^\star$ is lower bounded by an affine function of the perturbation parameters u and v. There are many consequences of this fact, for example, if $\lambda_i^\star$ is very large, then tighten the ith inequality constraint by $u_i < 0$ is guaranteed to increase the optimal value $q^\star(u, v)$ of the problem (6.69) largely.

Moreover, if the function $q^\star$ is differentiable at the point $(0, 0)$, then the inequality (6.71) implies that the gradient $\nabla q^\star(0, 0)$ is given by

$$\frac{\partial q^\star(0,0)}{\partial u_i} = -\lambda_i^\star, \quad i = 1, \ldots, m, \tag{6.72}$$

and

$$\frac{\partial q^\star(0,0)}{\partial v_i} = -\nu_i^\star, \quad i = 1, \ldots, p. \tag{6.73}$$

In other words, the dual optimal points $\lambda^\star$ and $\nu^\star$ of the problem (6.58) represent the *local sensitivity* of the optimal value of (6.58) with respect to perturbations on the constraints. When the original problem (6.58) is convex (and strong duality holds), we have the following intuition for why (6.72) and (6.73) hold: When the problem (6.58) is convex, then the function $q^\star$ is convex in (u, v) (see exercise 6.3), so the global underestimator (6.71) implies that the vector $\begin{bmatrix} -\lambda^{\star T} & -\nu^{\star T} \end{bmatrix}$ must be the gradient of $q^\star$ at the point $(0, 0)$. (See exercise 6.4 for a formal proof regarding this result.)

6.6.4 Optimality conditions

When strong duality holds for the problem (6.58), we can obtain some useful properties and conditions related to a primal optimal point $x^\star$ of (6.58) and a dual optimal point $(\lambda^\star, \nu^\star)$ of (6.65). Note that, still, unless mentioned explicitly, we do not make any assumptions about the convexity of the problem (6.58) in the following discussions.

Complementary slackness

Suppose strong duality holds for the problem (6.58), and let $x^\star$ and $(\lambda^\star, \nu^\star)$ be optimal points of the primal and dual problems, respectively, then we have

$$\begin{aligned} f_0(x^\star) = p^\star = d^\star &= g(\lambda^\star, \nu^\star) && (6.74)\\ &= \inf_{x \in \mathcal{D}} \left(f_0(x) + \sum_{i=1}^m \lambda_i^\star f_i(x) + \sum_{i=1}^p \nu_i^\star h_i(x) \right) && (6.75)\\ &\leq f_0(x^\star) + \sum_{i=1}^m \lambda_i^\star f_i(x^\star) + \sum_{i=1}^p \nu_i^\star h_i(x^\star) && (6.76)\\ &\leq f_0(x^\star), && (6.77) \end{aligned}$$

where $\mathcal{D} = \bigcap_{i=0}^m \mathbf{dom}\, f_i \cap \bigcap_{i=1}^p \mathbf{dom}\, h_i$ is the common domain of the functions in the problem (6.58). The equality (6.74) is due to the strong duality assumption; the equality (6.75) follows since it is the definition of the dual function; the inequality

(6.76) holds since $x^\star \in \mathcal{D}$; and the last inequality (6.77) follows from the fact that $x^\star$ is primal feasible and $(\lambda^\star, \nu^\star)$ is dual feasible, *i.e.*,

$$\lambda^\star \succeq 0, \qquad f_i(x^\star) \leq 0, \quad i = 1, \ldots, m, \qquad h_i(x^\star) = 0, \quad i = 1, \ldots, p.$$

Therefore, the inequalities (6.76) and (6.77) must hold with equality, which implies that:

- The point $x^\star$ is *a* minimizer of the Lagrangian $L(x, \lambda^\star, \nu^\star)$ (but not necessarily the unique minimizer), since the infimum in the definition of the dual function $g(\lambda^\star, \nu^\star)$ is attained at $x^\star$.
- The dual optimal point $\lambda^\star$ and the primal optimal point $x^\star$ must satisfy

  $$\sum_{i=1}^{m} \lambda_i^\star f_i(x^\star) = 0.$$

  Noticing that each term $\lambda_i^\star f_i(x^\star)$ in the above summation is nonpositive, we conclude that

  $$\lambda_i^\star f_i(x^\star) = 0, \quad i = 1, \ldots, m. \tag{6.78}$$

The equalities in (6.78) is called the *complementary slackness* condition, which says that for each $i = 1, \ldots, m$:

- if $\lambda_i^\star > 0$, then we must have $f_i(x^\star) = 0$, *i.e.*, the ith inequality constraint must be active, or tight, at the primal optimal point $x^\star$;
- if $f_i(x^\star) < 0$, then we must have $\lambda_i^\star = 0$, *i.e.*, the ith optimal Lagrange multiplier for the inequality constraints must be zero.

KKT optimality conditions

Now we assume that the functions $f_0, f_1, \ldots, f_m$ and $h_1, \ldots, h_p$ in the problem (6.58) are all differentiable (but not necessarily convex), and strong duality holds. According to the previous discussions, we have the following *necessary* conditions for $x^\star$ and $(\lambda^\star, \nu^\star)$ to be primal and dual optimal:

$$\begin{array}{rcll} f_i(x^\star) & \leq & 0, & i = 1, \ldots, m \\ h_i(x^\star) & = & 0, & i = 1, \ldots, p \\ \lambda_i^\star & \geq & 0, & i = 1, \ldots, m \\ \lambda_i^\star f_i(x^\star) & = & 0, & i = 1, \ldots, m \\ \nabla f_0(x^\star) + \sum_{i=1}^m \lambda_i^\star \nabla f_i(x^\star) + \sum_{i=1}^p \nu_i^\star \nabla h_i(x^\star) & = & 0, & \end{array} \tag{6.79}$$

which are called the *Karush-Kuhn-Tucker (KKT)* optimality conditions for the problem (6.58). The first and second conditions in (6.79) are the primal feasibility conditions, the third and fourth conditions are the dual feasibility and complementary slackness conditions, respectively, and the last condition follows from the fact that the gradient of the Lagrangian $L(x, \lambda^\star, \nu^\star)$ with respect to x must vanish at the primal optimal point $x^\star$. The KKT conditions (6.79) holds for *any* optimization

problem with differentiable objective and constraint functions for which strong duality obtains.

Besides strong duality and differentiability, if we additionally assume that the primal problem (6.58) is convex, *i.e.*, the functions $f_0, f_1, \ldots, f_m$ are convex and $h_1, \ldots, h_p$ are affine, then the KKT conditions (6.79) are not only necessary but also *sufficient* for optimality, which means that any point $\tilde{x}$ and $(\tilde{\lambda}, \tilde{\nu})$ that satisfy the KKT conditions must be primal and dual optimal, respectively.

Remark 6.6 *Proof of the KKT conditions for convex problems.* We show that if the primal problem (6.58) is convex and differentiable, then any point $\tilde{x}$ and $(\tilde{\lambda}, \tilde{\nu})$ that satisfy the KKT conditions

$$\begin{aligned} f_i(\tilde{x}) &\leq 0, \quad i = 1, \ldots, m \\ h_i(\tilde{x}) &= 0, \quad i = 1, \ldots, p \\ \tilde{\lambda}_i &\geq 0, \quad i = 1, \ldots, m \\ \tilde{\lambda}_i f_i(\tilde{x}) &= 0, \quad i = 1, \ldots, m \\ \nabla f_0(\tilde{x}) + \textstyle\sum_{i=1}^m \tilde{\lambda}_i \nabla f_i(\tilde{x}) + \sum_{i=1}^p \tilde{\nu}_i \nabla h_i(\tilde{x}) &= 0, \end{aligned} \tag{6.80}$$

are primal and dual optimal, respectively.

Firstly, notice that the first two conditions in (6.80) imply that $\tilde{x}$ is primal feasible, and the third condition implies that $(\tilde{\lambda}, \tilde{\nu})$ is dual feasible. When the primal problem is convex and $\tilde{\lambda} \succeq 0$, the Lagrangian

$$L(x, \tilde{\lambda}, \tilde{\nu}) = f_0(x) + \sum_{i=1}^m \tilde{\lambda}_i f_i(x) + \sum_{i=1}^p \tilde{\nu}_i h_i(x)$$

is a convex function in x, so the last condition implies that $\tilde{x}$ is a global minimizer of the Lagrangian $L(x, \tilde{\lambda}, \tilde{\nu})$. Hence, we have

$$g(\tilde{\lambda}, \tilde{\nu}) = L(\tilde{x}, \tilde{\lambda}, \tilde{\nu}) = f_0(\tilde{x}) + \sum_{i=1}^m \tilde{\lambda}_i f_i(\tilde{x}) + \sum_{i=1}^p \tilde{\nu}_i h_i(\tilde{x}) = f_0(\tilde{x}),$$

where the last equality follows from the feasibility of $\tilde{x}$ and the complementary slackness condition. Then by (6.68), the duality gap between the primal and dual feasible points $\tilde{x}$ and $(\tilde{\lambda}, \tilde{\nu})$ is zero, which implies that $\tilde{x}$ and $(\tilde{\lambda}, \tilde{\nu})$ must be primal and dual optimal, respectively.

Notice that in the above proof, we did not make any assumptions regarding the strong duality of the problem (6.58), so the KKT conditions are sufficient (but not necessary) for *any convex* optimization problem with differentiable objective and constraint functions.

6.7 Infeasible problems

Consider the following feasibility problem:

$$\begin{array}{lll} \text{find} & x & \\ \text{subject to} & f_i(x) \leq 0, & i = 1, \ldots, m \\ & h_i(x) = 0, & i = 1, \ldots, p \end{array} \tag{6.81}$$

with variable $x \in \mathbf{R}^n$. Suppose that this problem is infeasible, *i.e.*, there is no point x that satisfies all the constraints, then a natural question is how to find the largest subset of the constraints that can be satisfied simultaneously. In other words, we want to identify the smallest subset of the constraints in (6.81) that must be removed so that the remaining constraints are feasible.

If the total number of constraints $m + p$ is not too large, then we can simply enumerate all the subsets of the constraints (from the largest to the smallest) and check their feasibility, and quit as soon as we find a feasible subset. This approach needs to solve 2^{m+p} feasibility problems in the worst case, which is clearly not scalable when $m + p$ is large, *e.g.*, more than 50.

6.7.1 Relaxation and penalty heuristics

An effective heuristic for this problem is to introduce some auxiliary variables to measure the violation of the constraints, and then solve a penalized version of the original problem that minimizes the total constraint violation. Specifically, we introduce two auxiliary variables $u \in \mathbf{R}^m$ and $v \in \mathbf{R}^p$ to measure the violation of the inequality and equality constraints, respectively, and then solve the following optimization problem:

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^m u_i + \sum_{i=1}^p |v_i| \\ \text{subject to} & u \succeq 0, \quad f_i(x) \leq u_i, \quad i = 1, \ldots, m \\ & h_i(x) = v_i, \quad i = 1, \ldots, p \end{array} \tag{6.82}$$

with variables $x \in \mathbf{R}^n$, $u \in \mathbf{R}^m$, and $v \in \mathbf{R}^p$. The problem (6.82) is always feasible, since for any $x \in \mathbf{R}^n$, we can choose sufficiently large u and v to satisfy the constraints.

Let $(x^\star, u^\star, v^\star)$ be an optimal point of the problem (6.82), then the point $x^\star$ is expected to satisfy a large fraction of the constraints in (6.81), while the violation of the other constraints is given by the nonzero entries of $u^\star$ and $v^\star$.

If the original feasibility problem (6.81) is convex, then the heuristic given by (6.82) is also a convex problem, and hence can be solved much more efficiently than the combinatorial problem (6.83).

Intuition and interpretations

The intuition behind the heuristic (6.82) is that, its objective corresponds to minimizing the ℓ_1-norm of the vector $(u_1, \ldots, u_m, v_1, \ldots, v_p)$ of constraint violations, which hence encourages the solution $u^\star$ and $v^\star$ of the problem (6.82) to be sparse, so that the number of violated constraints is small.

This explanation is also seen by noticing that finding the largest feasible subset of the problem (6.81) is equivalent to

$$\begin{array}{ll} \text{minimize} & \mathbf{card}\, u + \mathbf{card}\, v \\ \text{subject to} & u \succeq 0, \quad f_i(x) \leq u_i, \quad i = 1, \ldots, m \\ & h_i(x) = v_i, \quad i = 1, \ldots, p, \end{array} \tag{6.83}$$

where $x \in \mathbf{R}^n$, $u \in \mathbf{R}^m$, and $v \in \mathbf{R}^p$ are the variables, and the cardinalities $\mathbf{card}\, u$ and $\mathbf{card}\, v$ denote the number of nonzero entries of u and v, respectively. Since we have required $u \succeq 0$, applying the ℓ_1-norm heuristic (see §5.2.3, page 161) on the cardinality functions in (6.83) gives us the problem (6.82).

6.7.2 Exact penalty method

Applying the previous idea to the general optimization problem (6.58) yields the *exact penalty method* for optimization, which is given by

$$\begin{array}{ll} \text{minimize} & f_0(x) + \gamma \left(\sum_{i=1}^m u_i + \sum_{i=1}^p |v_i|\right) \\ \text{subject to} & u \succeq 0, \quad f_i(x) \leq u_i, \quad i = 1, \ldots, m \\ & h_i(x) = v_i, \quad i = 1, \ldots, p \end{array}$$

with variables $x \in \mathbf{R}^n$, $u \in \mathbf{R}^m$, and $v \in \mathbf{R}^p$, where $\gamma > 0$ is a penalty parameter that controls the strength of the penalty on the constraint violation. This problem is sometimes also written in compact form as

$$\text{minimize} \quad f_0(x) + \gamma \left(\sum_{i=1}^m f_i(x)_+ + \sum_{i=1}^p |h_i(x)|\right), \tag{6.84}$$

where $f_i(x)_+ = \max\{f_i(x), 0\}$ is the positive part of $f_i(x)$, *i.e.*, the violation of the ith inequality constraint. The problem (6.84) is a convex optimization problem if the original problem (6.58) is convex.

Remark 6.7 We may provide several interpretations for the exact penalty method (6.84) for the problem (6.58).

Relaxation interpretation. It is easily seen that the problem (6.84) is a relaxation of the original problem (6.58), since it essentially ignores all the constraints in (6.58) and instead penalizes the total constraint violation in the objective. This simple relaxation has a very useful property at the optimal point: If an optimal point of the problem (6.84) is feasible for the original problem (6.58), then it is also an optimal point of the original problem. To see this, let $x^\star$ be an optimal point of the problem (6.84), then we have

$$f_0(x^\star) + \gamma \left(\sum_{i=1}^m f_i(x^\star)_+ + \sum_{i=1}^p |h_i(x^\star)|\right) \leq f_0(x) + \gamma \left(\sum_{i=1}^m f_i(x)_+ + \sum_{i=1}^p |h_i(x)|\right)$$

for all $x \in \mathbf{R}^n$ (that are, of course, in the domain of the objective function). Now suppose that $x^\star$ is feasible for the original problem (6.58), then for all x that are feasible for (6.58), the penalty terms in both sides of the above inequality are zero, so we have

$$f_0(x^\star) \leq f_0(x),$$

which implies that $x^\star$ is an optimal point of the original problem (6.58).

Regularization interpretation. We can also interpret the penalty term in the problem (6.84) as a regularizer that incorporates the prior knowledge that $f_i(x)$ should be non-positive and $h_i(x)$ should be close to zero, *i.e.*, the constraints in the original problem (6.58) should be approximately satisfied. Hence, applying the exact penalty method corresponds to transforming the hard constraints in the original problem (6.58) into

soft regularization terms in the problem (6.84). When the original problem is infeasible, *i.e.*, the prior knowledge implemented in the constraints cannot be satisfied simultaneously, we can only hope to find a point that is consistent with the prior knowledge as much as possible, where our irritation of the misalignment is controlled by the penalty parameter $\gamma > 0$. When a larger penalty parameter γ is used, the solution of the problem (6.84) is expected to satisfy the constraints more closely, while a smaller γ allows for more violation of the constraints.

If the original problem (6.58) is feasible, then with sufficiently large $\gamma > 0$, solving the exact penalty problem (6.84) will yield an optimal point $x^\star$ such that

$$f_i(x^\star) \leq 0, \quad i = 1, \ldots, m,$$

and

$$h_i(x^\star) = 0, \quad i = 1, \ldots, p,$$

i.e., the point $x^\star$ is feasible for (6.58), and hence is also an optimal point of the original problem. In other words, the problems (6.58) and (6.84) are equivalent if the penalty coefficient γ is sufficiently large.

On the other hand, if the original problem (6.58) is infeasible, then solving the problem (6.84) will yield an optimal point $x^\star$ that leads to a sparse violation of the constraints, which hence approximately identifies the largest feasible subset of the original problem (6.58).

These ideas are illustrated in the following example.

Example 6.15 *Infeasible least norm problem.* We consider the following least norm problem:

$$\begin{array}{ll} \text{minimize} & \|x\|_2^2 \\ \text{subject to} & Ax = b, \end{array} \tag{6.85}$$

where $x \in \mathbf{R}^n$ is the variable, and $A \in \mathbf{R}^{m \times n}$ and $b \in \mathbf{R}^m$ are given. Suppose that the problem is infeasible, *i.e.*, there is no point x that satisfies the equality constraint (which usually happens when $m > n$). Then we can apply the exact penalty method to find a point that is as close as possible to satisfying the equality constraint, *i.e.*, solving the problem

$$\text{minimize} \quad \|x\|_2^2 + \gamma \|Ax - b\|_1 \tag{6.86}$$

with variable $x \in \mathbf{R}^n$, where $\gamma > 0$ is a penalty parameter.

Figure 6.9 illustrates the tradeoff between the least norm objective $\|x\|_2^2$ and the constraint violation $\|Ax - b\|_1$ for different values of γ, under randomly generated problem data with $m = 100$ and $n = 10$. The horizontal dashed line in the figure represents the optimal value of the least norm problem (6.85) on the largest feasible subset obtained by combinatorial search. As γ increases, solution of the problem (6.86) has a smaller constraint violation, but results in a larger least norm objective. In particular, when γ is sufficiently large (around 0.1 in this example), the optimal point $x^\star$ of the problem (6.86) is the same as the solution of (6.85) on the largest feasible subset. At this point, the value $\|Ax^\star - b\|_1$ represents the smallest total violation of the equality constraint $Ax = b$ in (6.85).

Figure 6.10 plots the violation amplitude of each equality constraint $|a_i^T x^\star - b_i|$, $i = 1, \ldots, m$, for the optimal point $x^\star$ of the problem (6.86) with different values of γ,

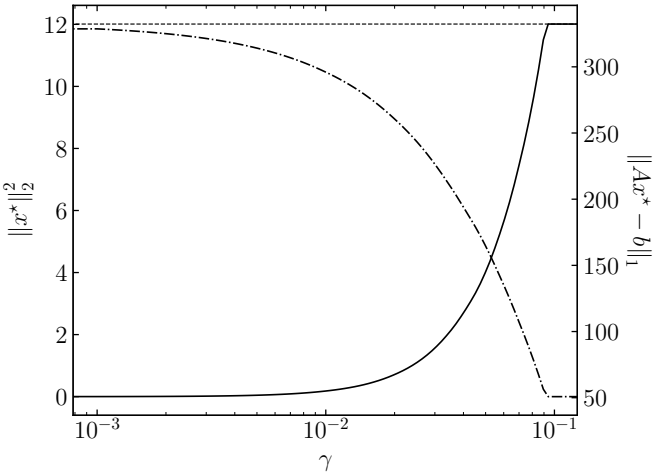

Figure 6.9 Least norm objective $\|x\|_2^2$ (shown solid, axis on the left) and constraint violation $\|Ax - b\|_1$ (shown dashedotted, axis on the right) for the optimal point $x^\star$ of the problem (6.86) with different values of γ. The dashed horizontal line represents the optimal value of the least norm problem (6.85) on the largest feasible subset obtained by combinatorial search.

where a_i^T is the ith row of the matrix A. The constraint violation for the solution of the problem (6.85) on the largest feasible subset is also shown darker for comparison. When γ is small, almost all the constraints $Ax = b$ are more or less violated. As γ increases, the constraint violation amplitudes become smaller, and eventually converge to the violation amplitudes under the solution of the problem (6.85) on the largest feasible subset, which is sparse and only violates a small fraction of the constraints.

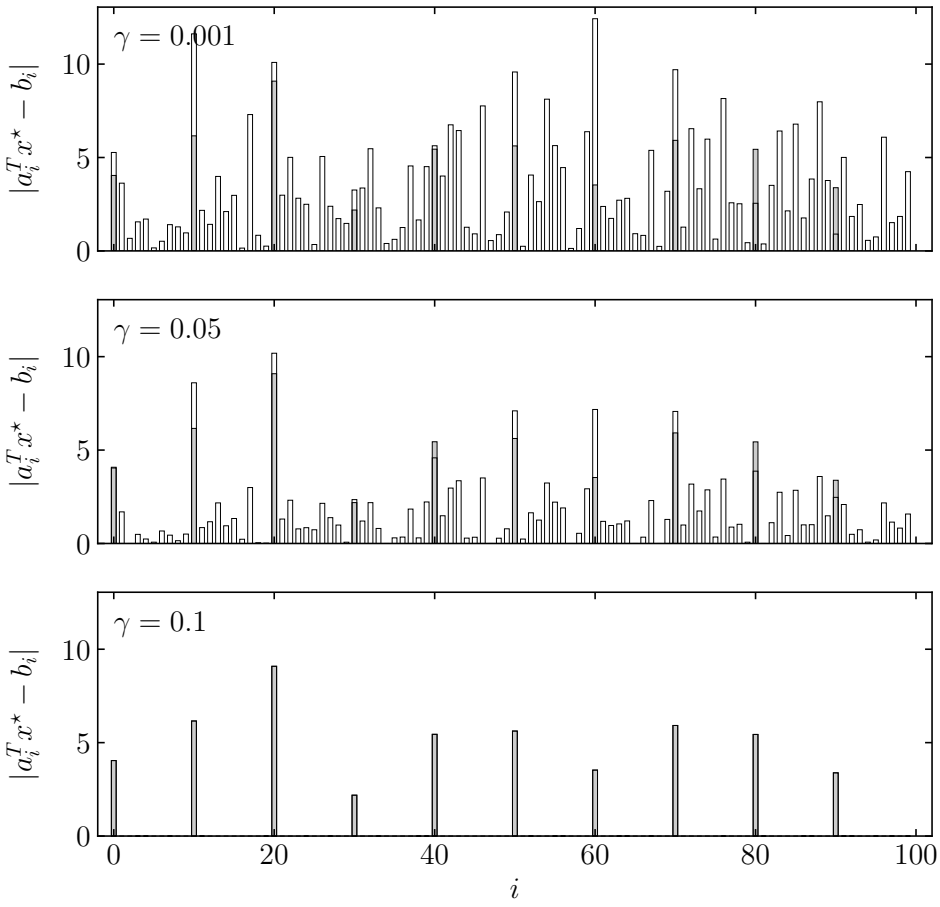

Figure 6.10 Violation amplitude of each equality constraint $|a_i^T x^\star - b_i|$, $i = 1, \ldots, m$, for the optimal point $x^\star$ of the problem (6.86) with different values of γ. The violation amplitudes for the solution of the problem (6.85) on the largest feasible subset are shown darker for reference. When $\gamma = 0.1$, the optimal point $x^\star$ of the problem (6.86) is the same as the solution of (6.85) on the largest feasible subset, and hence the two constraint violation amplitudes overlap exactly.

Bibliographical notes

Function fitting problems in the form (6.33) with an ℓ_1-norm regularization on the variable x are sometimes called *basis pursuit* problems; see Boyd and Vandenberghe [BV04, §6.5.4]. Some applications of basis pursuit in time series analysis and signal processing can be found in the review paper by Chen *et al.* [CDS01].

The boolean linear program (6.53) is a special case of the more general class of *integer programming* problems, which date back to the early 1950s [DFJ54, Gom58]. In fact, Kantorovich [Kan60] formulated a similar problem in the context of optimal resource allocation in the 1939 in the Soviet Union, but his work was largely unknown in the West until the late 1950s. Integer programming is NP-complete [Kar72, Pap81]. If some of the variables in an integer program are allowed to take continuous values, the problem is called a *mixed integer program* [Wol08, Bix12, AW13]. For more theoretical and practical aspects of integer programming, see, *e.g.*, the books by Schrijver [Sch98] and Wolsey [Wol21].

The two-way partitioning problem defined by (6.55) is also known as the *maximum cut* problem [Com09]. The semidefinite relaxation heuristic (6.57) for approximately solving the maximum cut problem was first proposed by Goemans and Williamson [GW95], which achieves the best known approximation ratio of 0.87856. Another heuristic for the maximum cut problem via the dual problem is discussed in Boyd and Vandenberghe [BV04, pages 219–220] and Wolkowicz and Zhao [WZ99].

Relaxations are also helpful in solving the inverse problem of *multi-armed bandits*; see Zhu *et al.* [ZHZB26]. Applications of semidefinite relaxation in other types of optimization problems, *e.g.*, nonconvex quadratic programming, can be found in Nesterov [Nes98], Zhang [Zha00], and Luo *et al.* [LMS+10].

Lagrangian relaxation and Lagrange duality are fundamental tools (both theoretically and numerically) in optimization. The name came from Lagrange's method of multipliers for solving equality constrained optimization problems appeared in mechanics, which was first proposed by Lagrange in 1788 [Lag53]. The related materials are covered in much more detail in many textbooks for optimization, *e.g.*, Luenberger [Lue69, chapter 8], Rockafellar [Roc70, part VI], Whittle [Whi71], Hiriart-Urruty and Lemaréchal [HL93a, chapter VII] and [HL93b, chapter XII], Bertsekas *et al.* [BNO03, chapters 5–7], Boyd and Vandenberghe [BV04, chapter 5], Borwein and Lewis [BL06], Nocedal and Wright [NW06, chapter 12], Bertsekas [Ber09, chapters 4 and 5], and Bertsekas [Ber16, chapters 4–7]; just list a few.

A simple constraint qualification condition that guarantees strong duality for convex problems is the so-called *Slater's condition*, which requires the existence of a strictly feasible point for the problem; see Boyd and Vandenberghe [BV04, §5.2.3] and Rockafellar [Roc70, theorem 28.2, page 277]. On rare occasions strong duality obtains for a *nonconvex* problem. An important example is that, strong duality holds for *any* optimization problem with (possibly nonconvex) quadratic objective and *one* quadratic inequality constraint, provided Slater's condition holds; see [BV04, appendix B] and the references therein. This result is sometimes called the *S-procedure* in control, and is very useful in trust region methods for approximately solving nonlinear optimization problems; see also §3.3 and [NW06, chapters 4 and 18].

The KKT conditions were originally named after Kuhn and Tucker (and hence it is sometimes referred to as the *Kuhn-Tucker conditions*), who first published the conditions in 1951 [KT51]. However, it was later discovered that the necessary conditions (6.79) has been stated in an unpublished master's thesis by Karush in 1939, which was later summa-

rized by Kuhn in 1976 [Kuh76]. A related optimality conditions for inequality constrained problems was also derived by John in 1948 [Joh85].

Some theoretical discussions about the exact penalty method (6.84) and its equivalence to the original problem (6.58) can be found in the textbooks by, *e.g.*, Bertsekas *et al.* [BNO03, §5.5 and §7.3] and Nocedal and Wright [NW06, chapters 15 and 17]. See also appendix C and the reference therein.

Exercises

6.1 *Gaussian covariance estimation.* Consider the problem of estimating the covariance matrix $X = \mathbf{E}\, yy^T \in \mathbf{S}^n_{++}$ of a zero mean multivariate Gaussian random vector $y \in \mathbf{R}^n$, given the dataset $\{y_1, \ldots, y_m\}$ of m independent samples of y. In §4.2.4, we have shown that this problem can be formulated as the maximum likelihood estimation problem given by

$$\text{minimize} \quad -\log\det S + \mathbf{tr}(SY),$$

where $S = X^{-1} \in \mathbf{S}^n_{++}$ is the optimization variable, and $Y = (1/m)\sum_{i=1}^m y_i y_i^T$ is the sample covariance matrix. Express the following prior information about the covariance matrix X as convex constraints on the variable S. You should formulate the constraints in the form of linear matrix inequalities.

(a) Lower and upper matrix bounds on X, *i.e.*, $L \preceq X \preceq U$ for some given matrices $L, U \in \mathbf{S}^n_{++}$.

(b) Condition number bound on X, *i.e.*, $\lambda_{\max}(X)/\lambda_{\min}(X) \leq \kappa$ for some given $\kappa > 1$, where $\lambda_{\max}(X)$ and $\lambda_{\min}(X)$ are the largest and smallest eigenvalues of X, respectively.

(c) Bounds on the variance of some linear functions of the random vector $y \in \mathbf{R}^n$, *i.e.*, $\mathbf{E}\,(c_i^T y)^2 \leq \alpha_i$ for some given vectors $c_1, \ldots, c_k \in \mathbf{R}^n$ and scalars $\alpha_1, \ldots, \alpha_k > 0$.
Hint. Use the fact that $u^T P^{-1} u \leq t$ is equivalent to the linear matrix inequality

$$\begin{bmatrix} P & u \\ u^T & t \end{bmatrix} \succeq 0$$

for any $P \in \mathbf{S}^n_{++}$, $u \in \mathbf{R}^n$, and $t \in \mathbf{R}$. (This is a special case of the *Schur complement* condition for positive semidefinite matrices; see §A.2.3.)

6.2 *Relaxation of convex problems.* Consider the convex program

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & x \in \mathcal{X} \end{array}$$

with variable $x \in \mathbf{R}^n$. Suppose that $x^\star \in \mathcal{X}$ is an optimal point of this problem. Show that if $x^\star$ is an *interior point* of the constraint set $\mathcal{X}$, then $x^\star$ is also an optimal point of the unconstrained problem

$$\text{minimize} \quad f_0(x)$$

with variable $x \in \mathbf{R}^n$.

6.3 *Optimal value of perturbed convex problems.* Consider the convex optimization problem

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \ldots, m \\ & Ax - b = 0 \end{array} \tag{6.87}$$

with variable $x \in \mathbf{R}^n$, where $f_0, \ldots, f_m \colon \mathbf{R}^n \to \mathbf{R}$ are convex functions and $A \in \mathbf{R}^{p \times n}$, $b \in \mathbf{R}^p$. The function $q^\star \colon \mathbf{R}^m \times \mathbf{R}^p \to \mathbf{R}$ that gives the optimal value of the perturbed problem of (6.87) is defined as

$$q^\star(u, v) = \inf \left\{ f_0(x) \;\middle|\; \begin{array}{l} x \in \bigcap_{i=0}^m \mathbf{dom}\, f_i \\ f_i(x) \leq u_i, \quad i = 1, \ldots, m \\ Ax - b = v \end{array} \right\},$$

where $u \in \mathbf{R}^m$ and $v \in \mathbf{R}^p$ are the perturbation parameters of the inequality and equality constraints, respectively. Show that the function $q^\star$ is a convex function of the perturbation parameters u and v.
Hint. Use the results from exercise 2.12.

6.4 *Local sensitivity via the dual problem.* Show that if the function $q^\star : \mathbf{R}^m \times \mathbf{R}^p \to \mathbf{R}$ given by (6.70) is differentiable at the point $(0, 0)$, and strong duality holds for the original problem (6.58), then the gradient of $q^\star$ at the point $(0, 0)$ is given by

$$\nabla q^\star(0,0) = \begin{bmatrix} -\lambda^{\star T} & -\nu^{\star T} \end{bmatrix},$$

i.e.,

$$\lambda_i^\star = -\frac{\partial q^\star(0,0)}{\partial u_i} \quad \text{and} \quad \nu_i^\star = -\frac{\partial q^\star(0,0)}{\partial v_i},$$

where $(\lambda^\star, \nu^\star)$ is an optimal point of the dual problem (6.65) associated with (6.58).

6.5 *Lower bounding a nonconvex QP.* Consider a nonconvex quadratic program that we have discussed in §3.3.3, which is given by

$$\begin{array}{ll} \text{minimize} & f(x) = (1/2)x^T P x + q^T x \\ \text{subject to} & \|x\|_\infty \leq 1 \end{array}$$

with variable $x \in \mathbf{R}^n$, where the matrix $P \in \mathbf{S}^n$ is symmetric by not positive semidefinite, so the objective function f is nonconvex.

(a) What is the Lagrangian of this problem?

(b) What is the dual function of this problem?

(c) What is the dual problem of this nonconvex QP?

(d) Randomly generate an instance of this nonconvex QP and solve the associated dual problem to obtain a lower bound on the optimal value of the original problem. Compare the obtained lower bound with some approximate solutions obtained from sequential convex approximation as in §3.3.3, what could you say about the quality of the obtained approximate solutions based on the lower bounds?

Part III

Applications

Chapter 7

Robust models

7.1 Stochastic optimization

We consider a special class of optimization problems where the objective function and the constraints involve some random variables. Specifically, suppose we are given a (deterministic) optimization problem with inequality constraints of the form

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \ldots, m, \end{array}$$

where $x \in \mathbf{R}^n$ is the optimization variable, and $f_0, f_1, \ldots, f_m \colon \mathbf{R}^n \to \mathbf{R}$ are the objective and inequality constraint functions. In a *stochastic optimization problem*, the functions f_i are extended to have the form $f_i(x, \omega)$, which depend on both the optimization variable $x \in \mathbf{R}^n$ and a random parameter $\omega \in \mathbf{R}^q$.

The random variable ω models the uncertainty or variation of the *parameters* of the function f_i, which can be due to various reasons such as random variation in implementation, measurement noise, or simply lack of knowledge about the true parameters. It is usually assumed that we do not know about the specific value of the random variable ω, but we are given its probability distribution. The goal of of stochastic optimization is to choose the optimization variable x, so that

- the constraints $f_i(x, \omega) \leq 0$ are satisfied, and
- the objective value $f_0(x, \omega)$ is small,

on average, or *with high probability*, where the former leads to *stochastic programming*, and the latter leads to the class of *chance constrained problems*.

7.1.1 Stochastic programming

A stochastic program is an optimization problem of the form

$$\begin{array}{ll} \text{minimize} & \mathbf{E}\, f_0(x, \omega) \\ \text{subject to} & \mathbf{E}\, f_i(x, \omega) \leq 0, \quad i = 1, \ldots, m \end{array} \tag{7.1}$$

with optimization variable $x \in \mathbf{R}^n$, where the expectation is taken with respect to the probability distribution of the random parameter ω. The constraints in the problem (7.1) are sometimes called the *stochastic constraints*.

If the functions f_i are convex in x for each fixed ω, then the problem (7.1) is a convex optimization problem, since the expectation preserves convexity. However, this problem is not always tractable, since it is usually very difficult to evaluate the objective and constraints. In only a few cases, the expectations in the stochastic program (7.1) has analytical expressions so the problem can be cast as a deterministic optimization problem. In the other cases, we need to solve this problem approximately.

Example 7.1 *Stochastic least squares.* Consider a least squares problem where the feature matrix $A \in \mathbf{R}^{m\times n}$ and the response vector $b \in \mathbf{R}^m$ are random. Our goal is to find a vector $x \in \mathbf{R}^n$ that minimizes the expected least squares cost $\|Ax-b\|_2^2$, *i.e.*, to solve the problem

$$\text{minimize} \quad \mathbf{E}\,\|Ax-b\|_2^2 \tag{7.2}$$

with variable $x \in \mathbf{R}^n$. Noticing that the cost function can be expanded as

$$\begin{aligned}\mathbf{E}\,\|Ax-b\|_2^2 &= \mathbf{E}\,(Ax-b)^T(Ax-b)\\ &= \mathbf{E}(x^TA^TAx-2b^TAx+b^Tb)\\ &= x^T(\mathbf{E}\,A^TA)x-2(\mathbf{E}\,b^TA)x+\mathbf{E}\,b^Tb,\end{aligned}$$

the stochastic least squares problem (7.2) is equivalent to the deterministic convex quadratic program

$$\text{minimize} \quad x^TPx-2q^Tx+r$$

with variable $x \in \mathbf{R}^n$, where the problem data $P \in \mathbf{S}_+^n$, $q \in \mathbf{R}^n$, and $r \in \mathbf{R}$ are given by

$$P = \mathbf{E}\,A^TA, \quad q = \mathbf{E}\,A^Tb, \quad r = \mathbf{E}\,b^Tb. \tag{7.3}$$

Hence, solving the stochastic least squares problem (7.2) only requires evaluating these second moments involving the random parameters A and b.

As a specific example, assume that only the matrix A is a random variable taking values in $\mathbf{R}^{m\times n}$ with mean $\mathbf{E}\,A = \bar{A}$, and assume the vector b is a deterministic vector in $\mathbf{R}^m$. In this case, we can express A as

$$A = \bar{A}+U,$$

where $U \in \mathbf{R}^{m\times n}$ is a random matrix with zero mean, and therefore the objective of the problem (7.2) is given by

$$\begin{aligned}\mathbf{E}\,\|Ax-b\|_2^2 &= \mathbf{E}\,(\bar{A}x-b+Ux)^T(\bar{A}x-b+Ux)\\ &= (\bar{A}x-b)^T(\bar{A}x-b)+2\,\mathbf{E}\,(\bar{A}x-b)^TUx+\mathbf{E}\,x^TU^TUx\\ &= \|\bar{A}x-b\|_2^2+x^T(\mathbf{E}\,U^TU)x.\end{aligned}$$

Let $P = \mathbf{E}\,U^TU$, then the stochastic least squares problem (7.2) under the current assumption is equivalent to the deterministic convex quadratic program

$$\text{minimize} \quad \|\bar{A}x-b\|_2^2+\|P^{1/2}x\|_2^2.$$

(This result is also obtained by directly evaluating the moments in (7.3).) The solution of this problem is given by

$$x^\star = (\bar{A}^T\bar{A} + P)^{-1}\bar{A}^T b. \tag{7.4}$$

(Here we assume that the matrix $\bar{A}^T\bar{A}+P$ is invertible, which is indeed the case, *e.g.*, in the example below.) If we further make the assumption that the random matrix $U \in \mathbf{R}^{m\times n}$ has IID Gaussian entries with mean zero and variance σ^2, then we have $P = m\sigma^2 I$, and the solution given by (7.4) reduces to

$$x^\star = (\bar{A}^T\bar{A} + m\sigma^2 I)^{-1}\bar{A}^T b, \tag{7.5}$$

which is essentially the solution of the Tikhonov regularization least squares problem given by (5.13) on page 159 with $\gamma = m\sigma^2$. Therefore, we can interpret the Tikhonov regularization as a way to obtain a robust solution to the least squares problem, by taking into account possible random variation in the feature matrix A.

Mean field approximation

One of the most trivial but quite useful approximation to the stochastic programming problem (7.1) is to solve its *mean field approximation* (or the *certainty equivalent problem*), which is given by

$$\begin{array}{ll} \text{minimize} & f_0(x, \mathbf{E}\,\omega) \\ \text{subject to} & f_i(x, \mathbf{E}\,\omega) \leq 0, \quad i = 1, \ldots, m, \end{array} \tag{7.6}$$

where $x \in \mathbf{R}^n$ is the optimization variable, and $\mathbf{E}\,\omega$ is the mean of the random parameter ω. Roughly speaking, the mean field approximation (7.6) corresponds to simply ignoring the variation in the parameters ω of the functions f_i in the problem (7.1), and treating the random parameters ω as if they are fixed to their mean value $\mathbf{E}\,\omega$.

If the functions $f_i : \mathbf{R}^n \times \mathbf{R}^q \to \mathbf{R}$ are convex in $x \in \mathbf{R}^n$ for each fixed $\omega \in \mathbf{R}^q$, then the problem (7.6) is a (deterministic) convex optimization problem, since it is obtained by fixing the random parameter ω to its mean value $\mathbf{E}\,\omega$. Additionally, if the functions f_i are convex in ω for each fixed x (where in this case, the functions f_i are biconvex in x and ω), then by Jensen's inequality, we have

$$f_i(x, \mathbf{E}\,\omega) \leq \mathbf{E}\, f_i(x, \omega) \tag{7.7}$$

for $i = 0, 1, \ldots, m$. According to the inequality (7.7), we have the following relationship between the original stochastic program (7.1) and its mean field approximation (7.6):

- Any feasible point of the original stochastic program (7.1) is also a feasible point of the mean field approximation (7.6). (This also implies that if the mean field approximation (7.6) is infeasible, then the original stochastic program (7.1) must be infeasible.)
- The optimal value of the mean field approximation provides an lower bound on the optimal value of the original stochastic program (7.1).

In other words, when the objective and constraint functions $f_i\colon \mathbf{R}^n \times \mathbf{R}^q \to \mathbf{R}$ of the original stochastic program (7.1) are all biconvex, then the mean field approximation (7.6) corresponds to a deterministic convex relaxation of the original stochastic program (7.1).

Example 7.2 *Mean field approximation of stochastic least squares.* We consider the stochastic least squares problem (7.2), where the feature matrix $A \in \mathbf{R}^{m\times n}$ is a random matrix, and the response vector $b \in \mathbf{R}^m$ is a deterministic vector. In particular, we assume that the random matrix A has the form $A = \bar{A} + U$, where $\bar{A} \in \mathbf{R}^{m\times n}$ is the mean of A, and $U \in \mathbf{R}^{m\times n}$ has IID Gaussian entries with zero mean. Then the resulting instance of the problem (7.2) has analytical solution given by (7.5). The mean field approximation of this stochastic least squares problem is given by

$$\text{minimize} \quad \|\bar{A}x - b\|_2^2$$

with variable $x \in \mathbf{R}^n$, which is simply a least squares problem with solution

$$x^{\text{mf}} = (\bar{A}^T\bar{A})^{-1}\bar{A}^T b.$$

(Assuming that $m \geq n$ and $\bar{A}$ has full rank.) Figure 7.1 shows the distribution of the objective value $\|Ax - b\|_2^2$ of this stochastic least squares problem on a randomly generated dataset, when the optimization variable x is set to the optimal point $x^\star$ given by (7.5) and the mean field approximate solution x^{mf}. The optimal value of this problem on this dataset is 19.1, while the mean field approximation lower bound is 17.9.

Finite event set

Another case where the stochastic programming problem (7.1) is tractable occurs when the random parameter $\omega \in \mathbf{R}^q$ takes only a finite number of values, *i.e.*,

$$\mathbf{prob}(\omega = \omega_i) = p_i, \quad i = 1, \ldots, k,$$

where $\omega_i \in \mathbf{R}^q$ and the vector $p \in \mathbf{R}^k$ with $p \succeq 0$ and $\mathbf{1}^T p = 1$ is the probability distribution over the finite set $\{\omega_1, \ldots, \omega_k\}$. Now the stochastic programming problem (7.1) reduces to

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^k p_i f_0(x, \omega_i) \\ \text{subject to} & \sum_{i=1}^k p_i f_j(x, \omega_i) \leq 0, \quad j = 1, \ldots, m \end{array} \tag{7.8}$$

with optimization variable $x \in \mathbf{R}^n$, which is a deterministic optimization problem and is convex if the functions $f_0, f_1, \ldots, f_m$ are convex in x.

Example 7.3 *Stochastic linear approximation.* We consider a stochastic linear approximation problem of the form

$$\text{minimize} \quad \mathbf{E}\,\|Ax - b\|, \tag{7.9}$$

where $x \in \mathbf{R}^n$ is the variable and $\|\cdot\|$ is a norm (on $\mathbf{R}^m$). We assume that the feature matrix $A \in \mathbf{R}^{m\times n}$ is random, and the response vector $b \in \mathbf{R}^m$ is deterministic.

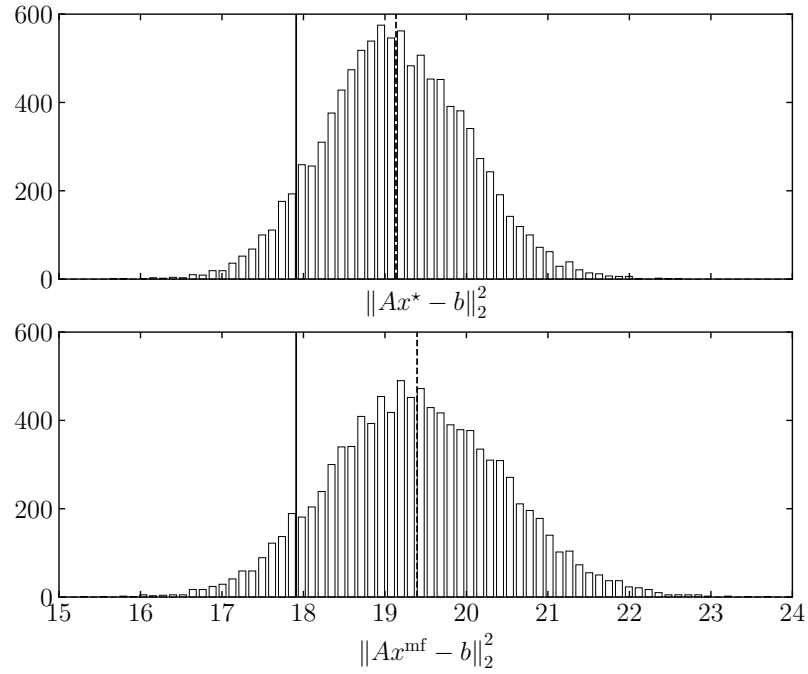

Figure 7.1 Distribution of the objective value $\|Ax - b\|_2^2$ of the stochastic least squares problem in example 7.2, under the optimal point $x^\star$ and the mean field approximate solution x^{mf}. The dashed lines illustrate the mean of the objective value distribution under the two solutions, and the solid lines in both plots display the lower bound given by the mean field approximation.

We have seen in example 7.1 that if the norm in (7.9) is the (squared) ℓ_2-norm, the problem is simply a convex quadratic program and has analytical solution. For the other norms, *e.g.*, the ℓ_1-norm and the ℓ_∞-norm, the problem (7.9) can be very difficult to solve. However, if the random matrix A is assumed to take values in a finite set $\{A_1, \ldots, A_k\}$ with probability distribution $p \in \mathbf{R}^k$, then the problem (7.9) reduces to

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^k p_i \|A_i x - b\| \end{array} \tag{7.10}$$

with variable $x \in \mathbf{R}^n$, which is often called a *sum-of-norms problem*.

If the ℓ_1-norm is used in the problem (7.10), then this problem can be expressed as

$$\begin{array}{ll} \text{minimize} & p^T t \\ \text{subject to} & \|A_i x - b\|_1 \leq t_i, \quad i = 1, \ldots, k, \end{array}$$

where $x \in \mathbf{R}^n$ and $t \in \mathbf{R}^k$ are the optimization variables. This problem is also equivalent to the linear program

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^k p_i \mathbf{1}^T u_i \\ \text{subject to} & -u_i \preceq A_i x - b \preceq u_i, \quad i = 1, \ldots, k, \end{array}$$

where $x \in \mathbf{R}^n$ and $u_i \in \mathbf{R}^m$ for $i = 1, \ldots, k$ are the optimization variables.

Similarly, if the ℓ_∞-norm is used in the problem (7.10), then we have the linear program

$$\begin{array}{ll} \text{minimize} & p^T t \\ \text{subject to} & -t_i \mathbf{1} \preceq A_i x - b \preceq t_i \mathbf{1}, \quad i = 1, \ldots, k, \end{array}$$

with variables $x \in \mathbf{R}^n$ and $t \in \mathbf{R}^k$.

Sample average approximation

A generic method for approximately solving the stochastic problem (7.1) is based on *Monte Carlo sampling*. The basic idea is to approximate the expectation in the problem (7.1) by a finite sample average, and then solve the resulting deterministic optimization problem. Specifically, we first draw N independent samples $\omega_1, \ldots, \omega_N$ from the probability distribution of the random parameter ω, and then consider the following *sample average approximation* to the original problem (7.1):

$$\begin{array}{ll} \text{minimize} & (1/N) \sum_{i=1}^N f_0(x, \omega_i) \\ \text{subject to} & (1/N) \sum_{i=1}^N f_j(x, \omega_i) \leq 0, \quad j = 1, \ldots, m \end{array} \tag{7.11}$$

with optimization variable $x \in \mathbf{R}^n$. The sample average approximation problem (7.11) can be considered as a special case of the finite event problem (7.8), where the probability distribution over the events is defined by $p \in \mathbf{R}^N$ with $p_i = 1/N$ for all $i = 1, \ldots, N$, and is hence much easier to solve.

It can be shown (under some technical conditions) that an optimal point x^{saa} of the sample average approximation problem (7.11) converges to the optimal point $x^\star$ of the original stochastic program (7.1) as $N \to \infty$. Moreover, the optimal value p^{saa} of the sample average approximation problem (7.11) provides a lower bound to the optimal value $p^\star$ of the original stochastic program (7.1), and this lower bound is tight as $N \to \infty$.

Example 7.4 *Stochastic ℓ_1-norm linear approximation.* Consider a stochastic linear approximation problem of the form (7.9) where the norm $\|\cdot\|$ in the objective is the ℓ_1-norm, *i.e.*,

$$\text{minimize} \quad \mathbf{E}\,\|Ax - b\|_1, \tag{7.12}$$

where both the feature matrix $A \in \mathbf{R}^{m\times n}$ and the response vector $b \in \mathbf{R}^m$ are random. We assume that the random parameter A has the form $A = \bar{A} + U$, where $\bar{A} \in \mathbf{R}^{m\times n}$ is the mean of A, and $U \in \mathbf{R}^{m\times n}$ has IID Gaussian entries with zero mean. Similarly, we assume that the random parameter b has the form $b = \bar{b} + v$, where $\bar{b} \in \mathbf{R}^m$ is the mean of b, and $v \in \mathbf{R}^m$ has IID Gaussian entries with zero mean.

Figure 7.2 shows the distribution of the objective value $\|Ax - b\|_1$ of the stochastic linear approximation problem (7.12) under the sample average approximation solution x^{saa} obtained by solving the problem

$$\text{minimize} \quad (1/N)\sum_{i=1}^{N} \|A_i x - b_i\|_1 \tag{7.13}$$

with $N = 10$ (shown top) and $N = 100$ (shown middle) samples, where A_i and b_i are the ith samples of the random parameters A and b, respectively. The histogram shown on the bottom corresponds to the mean field approximate solution x^{mf} according to the problem

$$\text{minimize} \quad \|\bar{A}x - \bar{b}\|_1. \tag{7.14}$$

The dashed lines in all three plots illustrate the mean of the objective value distribution under the respective solutions. The solid lines in all three plots are different lower bounds of the original problem (7.12) given by the respective approximations, *i.e.*, the optimal values of the problems (7.13) (with $N = 10$ and $N = 100$) and (7.14).

It is observed that, even with only $N = 10$ samples, the sample average approximation solution x^{saa} leads to an objective value distribution with lower mean, than the mean field approximate solution x^{mf}. This method also provides a much tighter lower bound on the optimal value of the original stochastic program (7.12), compared to the mean field approximation. When the number of samples increases to $N = 100$, the sample average approximation solution x^{saa} leads to an even better objective value distribution, and the lower bound given by the sample average approximation is also much tighter.

Note that in the previous example, the samples of the random parameters A and b used for solving the sample average approximation problem (7.13) are independent of the samples used for evaluating the distribution of $\|Ax - b\|_1$ shown in figure 7.2. In the most general case of stochastic programming, this actually provides a useful empirical criterion for choosing the number of samples N in the sample average approximation problem (7.11). Specifically, let x^{saa} be an optimal point of the sample average approximation problem (7.11) with N samples $\omega_1, \ldots, \omega_N$, and let

$$p^{\text{saa}} = (1/N)\sum_{i=1}^{N} f_0(x^{\text{saa}}, \omega_i)$$

be the optimal value of this problem. Then we can draw another independent set of N^{val} samples $\omega_1^{\text{val}}, \ldots, \omega_N^{\text{val}}$, and evaluate the quantity

$$p^{\text{val}} = (1/N^{\text{val}})\sum_{i=1}^{N^{\text{val}}} f_0(x^{\text{saa}}, \omega_i^{\text{val}}).$$

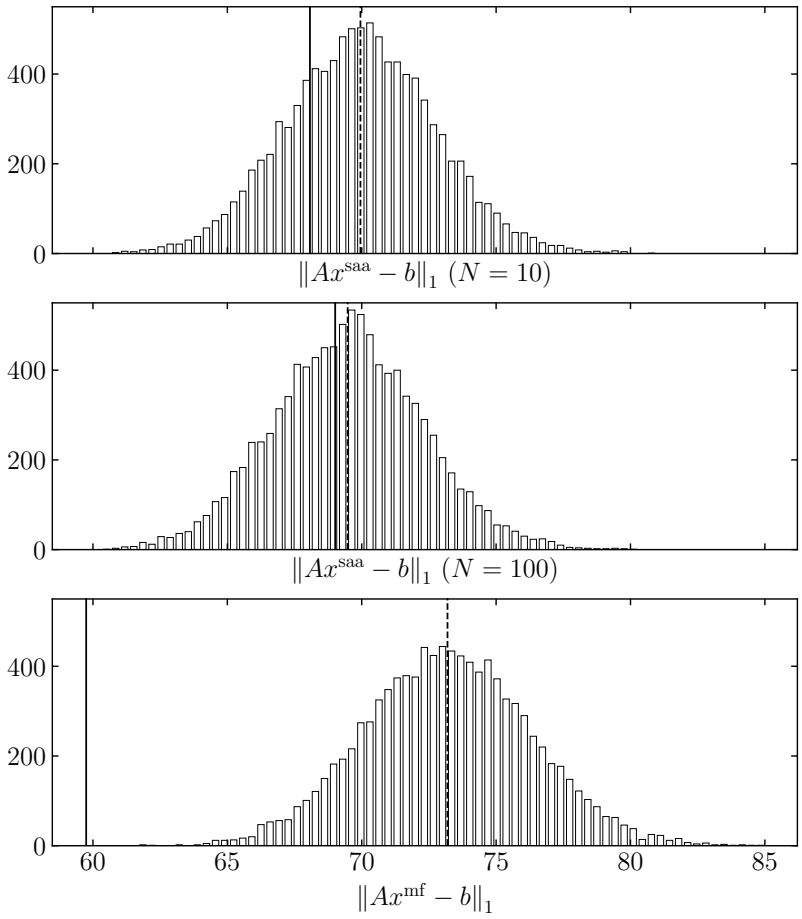

Figure 7.2 Distribution of the objective value $\|Ax - b\|_1$ of the problem (7.12) under the sample average approximation solution x^{saa} obtained by solving the problem (7.13) with $N = 10$ (shown top) and $N = 100$ (shown middle) samples, and the mean field approximate solution x^{mf} obtained by solving the problem (7.14) (shown bottom). The mean value of individual distribution is shown by the dashed lines, and the solid lines in all three plots illustrate the lower bounds given by the respective approximations.

If we observe that $p^{\mathrm{val}} \approx p^{\mathrm{saa}}$, then we can conclude that the sample average approximation solution x^{saa} is a good approximate solution to the original stochastic program (7.1), and the number of samples N used in the problem (7.11) is likely to be sufficient. Otherwise, we can increase the number of samples N in the problem (7.11) and repeat this procedure. This idea is sometimes called the *out-of-sample validation*, and is illustrated by the first two plots in figure 7.2.

7.1.2 Chance constrained problems

Let $f\colon \mathbf{R}^n \times \mathbf{R}^q \to \mathbf{R}$ be a function of the optimization variable $x \in \mathbf{R}^n$ and the random parameter $\omega \in \mathbf{R}^q$. A *chance constraint* is a constraint of the form

$$\mathbf{prob}(f(x,\omega) \leq 0) \geq \eta, \tag{7.15}$$

where $\eta \in (0,1)$ is a given *confidence level*. Typically, the confidence level η is chosen to be close to 1, *e.g.*, $\eta = 0.95$, so that the chance constraint requires that the constraint $f_i(x,\omega) \leq 0$ is satisfied with high probability. A *chance constrained problem* is then defined as an optimization problem with one or more chance constraints.

Example 7.5 *Quantile optimization.* The problem of finding a point $x \in \mathbf{R}^n$ so that a stochastic objective function $f_0(x,\omega)$ in the variable x is small with high probability (say, $\eta \in (0,1)$) can be formulated as a chance constrained problem. To do this, we introduce an auxiliary variable $t \in \mathbf{R}$ and consider the problem

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \mathbf{prob}(f_0(x,\omega) \leq t) \geq \eta, \end{array} \tag{7.16}$$

where the optimization variables are $x \in \mathbf{R}^n$ and $t \in \mathbf{R}$. A problem in the form (7.16) are sometimes called a *quantile optimization problem*, and the optimal value $t^\star$ of this problem corresponds to the η-*quantile* of the distribution of $f_0(x,\omega)$.

Chance constraints are typically nonconvex. One exception is when the function $f\colon \mathbf{R}^n \times \mathbf{R}^q \to \mathbf{R}$ is jointly convex in x and ω, and the random parameter ω has a *log-concave* distribution $p(\omega)$, *i.e.*, the function $\log p(\omega)$ is concave in ω. Then the resulting chance constraint (7.15) can be expressed as a convex constraint in x; see exercise 7.1. The example below shows another case where a chance constraint can be expressed as a convex constraint.

Example 7.6 *Linear inequality with Gaussian parameters.* Consider a chance constraint of the form

$$\mathbf{prob}(a^T x \leq b) \geq \eta, \tag{7.17}$$

where $x \in \mathbf{R}^n$ is the optimization variable, $a \in \mathbf{R}^n$ is a Gaussian random vector with mean $\bar{a}$ and covariance matrix Σ, and $b \in \mathbf{R}$ is a fixed scalar. We show that this chance constraint can be expressed as a convex constraint of x when $\eta \geq 0.5$.

To do this, first note that since $a \sim \mathcal{N}(\bar{a}, \Sigma)$, the mean and variance of the random variable $a^T x$ are given by

$$\mathbf{E}\, a^T x = \bar{a}^T x, \qquad \mathbf{var}\, a^T x = x^T \Sigma x,$$

and we conclude that $a^T x \sim \mathcal{N}(\bar{a}^T x, x^T \Sigma x)$ is also a Gaussian random variable. Noticing that the probability $\mathbf{prob}(a^T x \leq b)$ is equivalent to

$$\mathbf{prob}(a^T x \leq b) = \mathbf{prob}\left(\frac{a^T x - \bar{a}^T x}{\sqrt{x^T \Sigma x}} \leq \frac{b - \bar{a}^T x}{\sqrt{x^T \Sigma x}}\right),$$

and since $(a^T x - \bar{a}^T x)/\sqrt{x^T \Sigma x} \sim \mathcal{N}(0,1)$ is a standard Gaussian random variable, we have

$$\mathbf{prob}(a^T x \leq b) = \Phi\left(\frac{b - \bar{a}^T x}{\sqrt{x^T \Sigma x}}\right),$$

where Φ is the cumulative distribution function of a standard Gaussian random variable, given by

$$\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} e^{-t^2/2}\, dt.$$

Therefore, the chance constraint (7.17) is equivalent to

$$\Phi\left(\frac{b - \bar{a}^T x}{\sqrt{x^T \Sigma x}}\right) \geq \eta \qquad \Longleftrightarrow \qquad \frac{b - \bar{a}^T x}{\sqrt{x^T \Sigma x}} \geq \Phi^{-1}(\eta),$$

i.e.,

$$\bar{a}^T x - b + \Phi^{-1}(\eta) \|\Sigma^{1/2} x\|_2 \leq 0.$$

According to our assumption that $\eta \geq 0.5$, we have $\Phi^{-1}(\eta) \geq 0$, and hence this constraint is convex in x.

Conservative approximations

In the most general case, chance constrained problems are very difficult to solve. However, effective approximations exist in practice. Assume that the function $f\colon \mathbf{R}^n \times \mathbf{R}^q \to \mathbf{R}$ is convex in the optimization variable $x \in \mathbf{R}^n$ for each fixed random parameter $\omega \in \mathbf{R}^q$. Suppose $\phi\colon \mathbf{R} \to \mathbf{R}_+$ is a nonnegative, convex, and nondecreasing function, with

$$\phi(0) = 1.$$

Define the indicator function $g\colon \mathbf{R} \to \{0,1\}$ by

$$g(z) = \begin{cases} 1, & z > 0 \\ 0, & \text{otherwise.} \end{cases} \tag{7.18}$$

Then for any $\alpha > 0$, we have

$$\phi(z/\alpha) \geq g(z)$$

for all $z \in \mathbf{R}$, and hence

$$\mathbf{E}\, \phi(f(x,\omega)/\alpha) \geq \mathbf{E}\, g(f(x,\omega)) = \mathbf{prob}(f(x,\omega) > 0). \tag{7.19}$$

Therefore, the constraint

$$\mathbf{E}\, \phi(f(x,\omega)/\alpha) \leq 1 - \eta \tag{7.20}$$

implies that

$$\mathbf{prob}(f(x,\omega) > 0) \leq 1 - \eta \qquad \Longleftrightarrow \qquad \mathbf{prob}(f(x,\omega) \leq 0) \geq \eta,$$

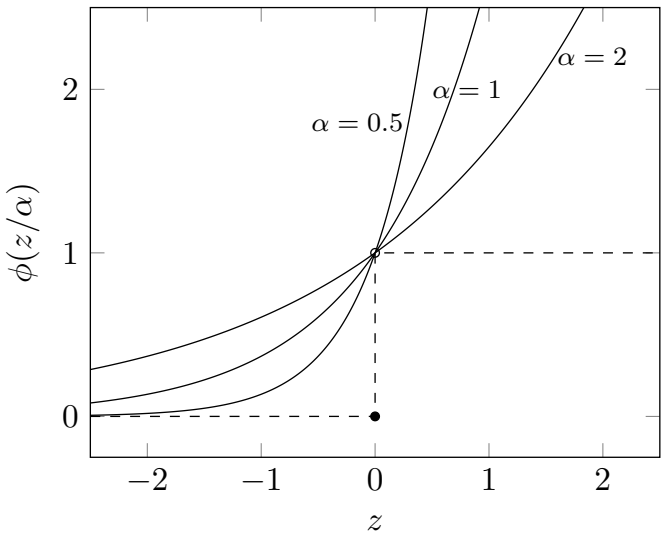

Figure 7.3 Approximation of the indicator function $g(z)$ given by (7.18) (shown dashed) with the exponential function $\phi(z/\alpha) = \exp(z/\alpha)$ (shown solid) for different values of α. When $\alpha = 0.5$, the function $\phi(z/\alpha)$ best approximates $g(z)$ for $z < 0$, while when $\alpha = 2$, the function $\phi(z/\alpha)$ best approximates $g(z)$ for $z > 0$.

i.e., the chance constraint (7.15) is satisfied. In other words, the constraint (7.20) provides a *conservative approximation* of the chance constraint (7.15), *i.e.*, it tightens the feasible set given by (7.15). Note that since ϕ is convex and nondecreasing, and f is convex in x, the composition $\phi(f(x,\omega)/\alpha)$ is convex in x (for each fixed $\alpha > 0$). As a result, the stochastic constraint (7.20) is a convex constraint of the optimization variable x.

Remark 7.1 *Interpretation and tightness.* By (7.19), we see that the approximation (7.20) of the original chance constraint (7.15) corresponds to replacing the zero-one indicator function $g(z)$ given by (7.18), applied to $z = f(x,\omega)$, with the convex nondecreasing function $\phi(z/\alpha)$. Figure 7.3 shows an example of this approximation with $\phi(z/\alpha) = \exp(z/\alpha)$ being the exponential function, for different values of α. (We will see later that this choice of ϕ corresponds to the *Chernoff bound* for approximating the chance constraint.)

Obviously, as the value of $\alpha > 0$ decreases, the approximation $\phi(z/\alpha)$ becomes tighter to the indicator function $g(z)$ for $z < 0$, but becomes looser for $z > 0$. (This is also illustrated in figure 7.3.) Therefore, if for all $\omega \in \mathbf{R}^q$, we have $f(x,\omega) \leq 0$ (or $f(x,\omega) \geq 0$), then as $\alpha \to 0$ (or $\alpha \to \infty$), the approximation (7.20) becomes tight to the original chance constraint (7.15) (while in these cases the chance constraint (7.15) does not really make sense in practice). In the most general case, however, the tightness of the approximation (7.20) to the original chance constraint (7.15) is *not* monotone in the parameter α, and depends on the distribution of $f(x,\omega)$.

Applying the approximation (7.20) in practice requires tuning the parameter $\alpha > 0$ so that the approximation is as tight as possible to the original chance constraint (7.15) (or at least includes the optimal point of the targeted chance constrained problem). However, according to the discussions in remark 7.1, this is not a trivial task. In fact, the constraint (7.20) can be expressed as a convex constraint *jointly* in x and α, so that they can be optimized together. In this case, the optimal value of α corresponding to the tightest possible approximation (7.20) of the original chance constraint (7.15) could be automatically determined by the optimization solver. To see this, we write (7.20) as

$$\mathbf{E}\,\alpha\phi(f(x,\omega)/\alpha) \leq \alpha(1-\eta). \tag{7.21}$$

Since the function $\alpha\phi(z/\alpha)$ (in the variable (z,α)) is the *perspective* of the convex nondecreasing function $\phi(z)$ (see exercise 2.13), it is jointly convex in z and α, and nondecreasing in z. As a result, the composition $\alpha\phi(f(x,\omega)/\alpha)$ is jointly convex in x and α for each fixed ω, and hence the expectation of this composition, *i.e.*, the left-hand side of (7.21), is also jointly convex in x and α. Therefore, by replacing the chance constraint (7.15) with (7.21), we obtain a convex conservative approximation which can be optimized over $x \in \mathbf{R}^n$ *and* $\alpha > 0$ simultaneously.

The following examples present some specific choices of the function ϕ that lead to different approximations of the chance constraint (7.15).

Example 7.7 *Markov bound.* Taking the function $\phi\colon \mathbf{R} \to \mathbf{R}_+$ in the approximation (7.19) as

$$\phi(z) = (z+1)_+ = \max\{0, z+1\}$$

gives the *Markov bound* of chance constraints for all $\alpha > 0$, which states that

$$\mathbf{prob}(f(x,\omega) > 0) \leq \mathbf{E}\,(f(x,\omega)/\alpha + 1)_+. \tag{7.22}$$

In this case, the approximation (7.21) of the chance constraint (7.15) becomes

$$\mathbf{E}\,(f(x,\omega) + \alpha)_+ \leq \alpha(1-\eta), \tag{7.23}$$

which is readily seen as a convex constraint in (x,α).

We give a basic example of applying the Markov approximation (7.23) to a chance constrained least norm problem. Specifically, we consider the problem

$$\begin{array}{ll} \text{minimize} & \|x\|_2 \\ \text{subject to} & \mathbf{prob}(f(Ax-b) \leq 0) \geq \eta, \end{array} \tag{7.24}$$

where $x \in \mathbf{R}^n$ is the variable, $A \in \mathbf{R}^{m\times n}$ is a random matrix, $b \in \mathbf{R}^m$ is a random vector, and the function $f\colon \mathbf{R}^m \to \mathbf{R}$ is given by

$$f(z) = \max_{i=1,\ldots,m} z_i, \tag{7.25}$$

which is a convex function of $z \in \mathbf{R}^m$. The Markov approximation of this problem is given by

$$\begin{array}{ll} \text{minimize} & \|x\|_2 \\ \text{subject to} & \mathbf{E}\,(f(Ax-b)+\alpha)_+ \leq \alpha(1-\eta), \end{array} \tag{7.26}$$

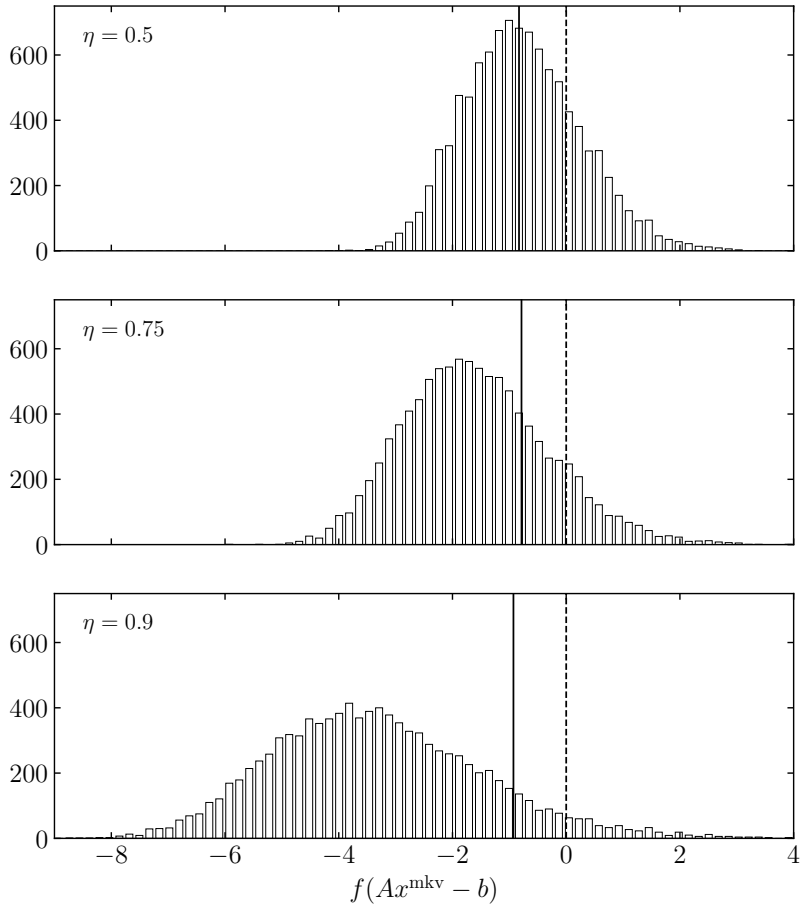

Figure 7.4 Distribution of the constraint function $f(Ax-b)$ (with f given by (7.25)) under the approximate solution x^{mkv} obtained by solving the problem (7.27), for $\eta = 0.5$, 0.75, and 0.9. The solid lines in the histogram are the η-quantiles of the distribution of $f(Ax^{\mathrm{mkv}}-b)$, corresponding to the thresholds for satisfying the chance constraint in the original problem (7.24).

which is a convex stochastic optimization problem with variables $x \in \mathbf{R}^n$ and $\alpha > 0$. The problem (7.26) can then be (approximately) solved by the sample average approximation method described on page 252. In particular, we consider the following convex program:

$$\begin{array}{ll} \text{minimize} & \|x\|_2 \\ \text{subject to} & (1/N)\sum_{i=1}^{N} \left(f(A_i x - b_i) + \alpha\right)_+ \leq \alpha(1-\eta), \end{array} \tag{7.27}$$

where A_i and b_i are the ith samples of the random parameters A and b, respectively, and the function f is given by (7.25). Figure 7.4 shows the distribution of the constraint function $f(Ax-b)$ under the approximate solution x^{mkv} obtained by solving the problem (7.27) with $N = 1000$ samples, for $\eta = 0.5$, 0.75, and 0.9. Note that, again, the samples of the random parameters A and b used for solving the problem (7.27) are independent of the samples used for evaluating the distribution of $f(Ax^{\mathrm{mkv}}-b)$ shown in the histogram. The solid lines in the histogram are the η-quantiles of the distribution of $f(Ax^{\mathrm{mkv}} - b)$, which are the thresholds for satisfying the chance constraint in the original problem (7.24). Empirically, since the η-quantile of the $f(Ax^{\mathrm{mkv}} - b)$ distribution is much below zero, the best point x^{mkv} that can be obtained under the Markov approximation (7.23) for this example is a *strict* feasible point of the original chance constraint in (7.24). In other words, here the Markov approximation (7.23) strictly tightens the original chance constraint in (7.24).

Example 7.8 *Chebyshev and Chernoff bounds.* Taking the function $\phi\colon \mathbf{R} \to \mathbf{R}_+$ in the approximation (7.19) as

$$\phi(z) = (z+1)_+^2 = \max\{0, z+1\}^2$$

gives the *Chebyshev bound* for all $\alpha > 0$, which states that

$$\mathbf{prob}(f(x,\omega) > 0) \leq \mathbf{E}\,(f(x,\omega)/\alpha + 1)_+^2, \tag{7.28}$$

and the approximation (7.21) of the chance constraint (7.15) becomes

$$\mathbf{E}\,(f(x,\omega) + \alpha)_+^2/\alpha \leq \alpha(1-\eta).$$

Moreover, taking ϕ as

$$\phi(z) = e^z$$

leads to the *Chernoff bound* for all $\alpha > 0$, given by

$$\mathbf{prob}(f(x,\omega) > 0) \leq \mathbf{E}\exp(f(x,\omega)/\alpha). \tag{7.29}$$

In this case, the approximation (7.21) of the chance constraint (7.15) becomes

$$\mathbf{E}\,\alpha\exp(f(x,\omega)/\alpha) \leq \alpha(1-\eta).$$

Figure 7.3 shows the approximation of $\phi(z/\alpha) = \exp(z/\alpha)$ to the indicator function $g(z)$ given by (7.18), for different values of α, and figure 7.5 compares the chosen functions ϕ corresponding to the Markov, Chebyshev, and Chernoff bounds.

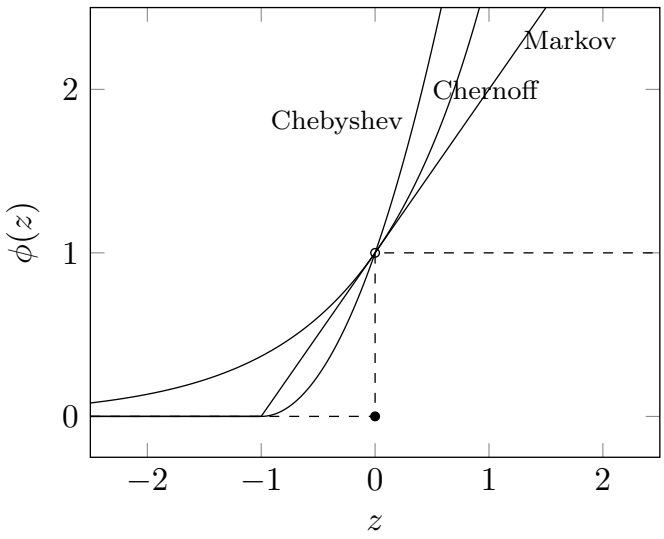

Figure 7.5 Approximation of the indicator function $g(z)$ given by (7.18) (shown dashed) with different functions $\phi(z)$ (shown solid) for $\alpha = 1$, corresponding to the Markov bound (7.22), Chebyshev bound (7.28), and Chernoff bound (7.29).

7.2 Worst-case robustness

Another approach of modeling uncertainty in the objective and constraint functions of an optimization problem is to consider the *worst-case* performance of the optimization variables under all possible realizations of the function parameters.

Specifically, let $f_0, f_1, \ldots, f_m \colon \mathbf{R}^n \times \mathbf{R}^q \to \mathbf{R}$ be functions of the optimization variable $x \in \mathbf{R}^n$ and the unknown parameter $\omega \in \mathbf{R}^q$, and we are given a set $U \subseteq \mathbf{R}^q$ that contains all possible realizations of the unknown parameter ω. Consider the following optimization problem:

$$\begin{array}{ll} \text{minimize} & f_0(x, \omega) \\ \text{subject to} & f_i(x, \omega) \leq 0, \quad i = 1, \ldots, m, \end{array} \tag{7.30}$$

which is parameterized by $\omega \in U$, and $x \in \mathbf{R}^n$ is the optimization variable. To take into account the worst-case robustness, rather than just solve the problem (7.30) for a specific value of $\omega \in U$, we want to find a solution that is robust under all possible values of ω in the set U, *i.e.*, to find a point $x \in \mathbf{R}^n$ such that *whatever* the actual realization of the parameter $\omega \in U$ in the problem (7.30) is, we always achieve the following:

- The objective $f_0(x, \omega)$ is minimized, and
- all the constraints $f_i(x, \omega) \leq 0$ for $i = 1, \ldots, m$ are satisfied.

7.2.1 Worst-case optimization

A solution of the problem (7.30) (that is robust under all $\omega \in U$) can be formally expressed as

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & f_0(x,\omega) \leq t \text{ for all } \omega \in U \\ & f_i(x,\omega) \leq 0 \text{ for all } \omega \in U, \quad i = 1,\ldots,m, \end{array} \tag{7.31}$$

where $x \in \mathbf{R}^n$ and $t \in \mathbf{R}$ are the optimization variables. This problem can be expressed more compactly as

$$\begin{array}{ll} \text{minimize} & \sup_{\omega \in U} f_0(x,\omega) \\ \text{subject to} & \sup_{\omega \in U} f_i(x,\omega) \leq 0, \quad i = 1,\ldots,m, \end{array} \tag{7.32}$$

where the optimization variable is $x \in \mathbf{R}^n$. The problem (7.32) is called a *worst-case optimization problem*, and is sometimes referred to as the *robust counterpart* of the original problem (7.30).

If the functions $f_0, f_1, \ldots, f_m$ are convex in x for each fixed ω, then we have the objective and constraint functions $\sup_\omega f_i(x,\omega)$ being convex in x, and hence the problem (7.32) is a convex optimization problem. However, similar to the stochastic optimization problem (7.1), the problem (7.32) is often difficult to solve in practice, since generally the functions $\sup_\omega f_i(x,\omega)$ do not have an analytical expression so it will be difficult to evaluate their values and gradients.

The problem (7.32) is tractable in some special cases. As a simple example, if the uncertainty set $U \subseteq \mathbf{R}^q$ is a finite set, then the worst-case problem (7.32) can be expressed as a problem with a finite number of constraints. In particular, let $U = \{\omega_1, \ldots, \omega_k\}$ be a finite set of k points in $\mathbf{R}^q$. Then the problem (7.32) reduces to

$$\begin{array}{ll} \text{minimize} & \max_{i=1,\ldots,k} f_0(x,\omega_i) \\ \text{subject to} & \max_{i=1,\ldots,k} f_j(x,\omega_i) \leq 0, \quad j = 1,\ldots,m \end{array}$$

with variable $x \in \mathbf{R}^n$, which can also be expressed in the form of (7.31) as

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & f_0(x,\omega_i) \leq t, \quad i = 1,\ldots,k \\ & f_j(x,\omega_i) \leq 0, \quad i = 1,\ldots,k, \;\; j = 1,\ldots,m \end{array}$$

with variables $x \in \mathbf{R}^n$ and $t \in \mathbf{R}$. If the functions $f_0, f_1, \ldots, f_m$ are convex in x for each fixed ω, then the problem above is a convex optimization problem with $k(m+1)$ constraints, and can therefore be solved efficiently. We provide more examples of tractable worst-case optimization problems in the next several paragraphs.

Robust linear programming

Consider an optimization problem of the form

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & \sup_{a_i \in \mathcal{A}_i} a_i^T x \leq b_i, \quad i = 1,\ldots,m, \end{array} \tag{7.33}$$

where $x \in \mathbf{R}^n$ is the optimization variable. We assume that only the coefficients $a_i \in \mathbf{R}^n$ in the linear constraints are uncertain and belong to the corresponding given uncertainty sets $\mathcal{A}_i \subseteq \mathbf{R}^n$, $i = 1, \ldots, m$. The vectors $c \in \mathbf{R}^n$ and $b \in \mathbf{R}^m$ are given and fixed. The problem (7.33) is called a *robust linear program*, and there are many forms of the uncertainty sets $\mathcal{A}_i$ that will lead to tractable robust linear programs.

Example 7.9 *Box uncertainty.* Suppose the uncertainty sets $\mathcal{A}_i$ in (7.33) are boxes, *i.e.*, of the form

$$\mathcal{A}_i = \{a \in \mathbf{R}^n \mid \bar{a}_i - v_i \preceq a \preceq \bar{a}_i + v_i\}, \quad i = 1, \ldots, m,$$

where $\bar{a}_i \in \mathbf{R}^n$ and $v_i \in \mathbf{R}^n_+$ are given. When $v_i = 0$, then it means that there is no uncertainty in the ith inequality constraint.

Given this uncertainty structure, to evaluate the supremum in the ith constraint of (7.33), first notice that

$$\sup_{a_i \in \mathcal{A}_i} a_i^T x = \sup_{a_i \in \mathcal{A}_i} \sum_{j=1}^n a_{ij} x_j = \sum_{j=1}^n \sup\{a_{ij} x_j \mid \bar{a}_{ij} - v_{ij} \leq a_{ij} \leq \bar{a}_{ij} + v_{ij}\}.$$

If $x_j \geq 0$, then the supremum of $a_{ij} x_j$ is attained with $a_{ij} = \bar{a}_{ij} + v_{ij}$, while if $x_j < 0$, then the supremum is attained with $a_{ij} = \bar{a}_{ij} - v_{ij}$. Therefore, with slight abuse of notation, letting

$$|x| = (|x_1|, \ldots, |x_n|)$$

be the vector of the absolute values of the entries of x, we have

$$\sup_{a_i \in \mathcal{A}_i} a_i^T x = \bar{a}_i^T x + v_i^T |x|.$$

With this, we could define the matrices $\bar{A} \in \mathbf{R}^{m \times n}$ and $V \in \mathbf{R}^{m \times n}_+$ by stacking the vectors $\bar{a}_i^T$ and v_i^T as rows, respectively, then the problem (7.33) can be expressed as

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & \bar{A} x + V|x| \preceq b, \end{array} \tag{7.34}$$

where $x \in \mathbf{R}^n$ is the variable.

The problem (7.34) is equivalent to the following linear program:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & \bar{A} x + V z \preceq b \\ & -z \preceq x \preceq z \end{array} \tag{7.35}$$

with variables $x \in \mathbf{R}^n$ and $z \in \mathbf{R}^n$. To see this, let (x, z) be a feasible point of the problem (7.35), then since $-z \preceq x \preceq z$, we have $|x| \preceq z$. Noticing that $V \in \mathbf{R}^{m \times n}_+$ is componentwise nonnegative, we have $V|x| \preceq Vz$, and hence

$$\bar{A} x + V|x| \preceq \bar{A} x + V z \preceq b,$$

which means that x is a feasible point of the problem (7.34). Conversely, if x is a feasible point of the problem (7.34), then by letting $z = |x|$, we have (x, z) being a feasible point of the problem (7.35).

Example 7.10 *Ellipsoidal uncertainty.* Suppose the coefficients a_i in (7.33) are known to lie within some ellipsoids, *i.e.*,

$$a_i \in \mathcal{A}_i = \{\bar{a}_i + P_i u \mid \|u\|_2 \leq 1\}, \quad i = 1, \ldots, m,$$

where $\bar{a}_i \in \mathbf{R}^n$ and $P_i \in \mathbf{R}^{n\times n}$ are given. If P_i is singular, then the ellipsoid $\mathcal{A}_i$ is degenerate (or 'flat') with dimension $\mathbf{rank}\, P_i$. If $P_i = 0$, then there is no uncertainty in the ith inequality constraint.

Under this ellipsoidal uncertainty set, the supremum in the ith constraint of (7.33) can be evaluated as

$$\begin{array}{rcl} \sup_{a_i\in\mathcal{A}_i} a_i^T x & = & \bar{a}_i^T x + \sup\{u^T P_i^T x \mid \|u\|_2 \leq 1\} \\ & = & \bar{a}_i^T x + \|P_i^T x\|_2, \end{array}$$

where the second equality follows from the definition of the *dual norm* (see §A.3.2, page 354). Therefore, the problem (7.33) can be expressed as

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & \bar{a}_i^T x + \|P_i^T x\|_2 \leq b_i, \quad i = 1, \ldots, m, \end{array} \tag{7.36}$$

where $x \in \mathbf{R}^n$ is the variable, which is a convex optimization problem with linear objective and convex inequality constraints.

Example 7.11 *Polyhedral uncertainty.* Suppose the coefficients a_i in (7.33) are known to lie within some polyhedra, *i.e.*,

$$a_i \in \mathcal{A}_i = \{a \in \mathbf{R}^n \mid C_i a \preceq d_i\}, \quad i = 1, \ldots, m,$$

where $C_i \in \mathbf{R}^{p_i\times n}$ and $d_i \in \mathbf{R}^{p_i}$ are given, and all the polyhedra $\mathcal{A}_i$ are assumed to be nonempty.

In this case, the problem (7.33) can be expressed as the following *bilevel optimization problem*:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & f_i(x) \leq b_i, \quad i = 1, \ldots, m, \end{array} \tag{7.37}$$

where $x \in \mathbf{R}^n$ is the variable, and the function $f_i : \mathbf{R}^n \to \mathbf{R}$ is defined as the optimal value of the following linear program:

$$\begin{array}{ll} \text{maximize} & x^T a \\ \text{subject to} & C_i a \preceq d_i, \end{array} \tag{7.38}$$

where $a \in \mathbf{R}^n$ is the variable, and x can be considered as a *parameter* of this linear program. In this context, the problem (7.37) is sometimes called the *outer problem*, and the problem (7.38) is called the (ith) *inner problem*. The Lagrangian associated with the inner problem (7.38) (in minimization form) is

$$L(a, \lambda) = -x^T a + \lambda^T (C_i a - d_i),$$

where $\lambda \in \mathbf{R}^{p_i}$ is the dual variable, and the corresponding dual function is

$$g(\lambda) = \inf_a L(a, \lambda) = \inf_a \left((C_i^T \lambda - x)^T a - \lambda^T d_i \right) = \begin{cases} -\lambda^T d_i, & C_i^T \lambda = x \\ -\infty, & \text{otherwise.} \end{cases}$$

Therefore, the dual problem of the inner problem (7.38) is given by

$$\begin{array}{ll} \text{minimize} & d_i^T \lambda \\ \text{subject to} & C_i^T \lambda = x \\ & \lambda \succeq 0, \end{array} \tag{7.39}$$

where $\lambda \in \mathbf{R}^{p_i}$ is the variable. Since the inner problem (7.38) is a linear program, we have strong duality, and hence $f_i(x)$ is also equal to the optimal value of the dual problem (7.39). As a result, we have $f_i(x) \leq b_i$ if and only if there exists $\lambda_i \in \mathbf{R}^{p_i}$ such that

$$d_i^T \lambda_i \leq b_i, \qquad C_i^T \lambda_i = x, \qquad \lambda_i \succeq 0,$$

for $i = 1, \ldots, m$. Therefore, the problem (7.37) can be expressed as the following linear program:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & d_i^T \lambda_i \leq b_i, \quad i = 1, \ldots, m \\ & C_i^T \lambda_i = x, \quad i = 1, \ldots, m \\ & \lambda_i \succeq 0, \quad i = 1, \ldots, m \end{array}$$

with variables $x \in \mathbf{R}^n$ and $\lambda_i \in \mathbf{R}^{p_i}$ for $i = 1, \ldots, m$.

Robust quadratic programming

Consider an optimization problem of the form

$$\begin{array}{ll} \text{minimize} & \sup_{P \in \mathcal{P}}((1/2)x^T P x + q^T x + r) \\ \text{subject to} & Ax \preceq b, \end{array} \tag{7.40}$$

where $x \in \mathbf{R}^n$ is the variable. We assume that only the quadratic coefficient matrix $P \in \mathbf{R}^{n \times n}$ in the objective is uncertain and belongs to a given uncertainty set $\mathcal{P} \subseteq \mathbf{R}^{n \times n}$, while the linear coefficient $q \in \mathbf{R}^n$ and the constant term $r \in \mathbf{R}$ are given and fixed. The problem (7.40) is sometimes called a *robust quadratic program*. We discuss some special cases of the uncertainty set $\mathcal{P}$ that will lead to tractable robust quadratic programs.

Example 7.12 *Finite set of matrices.* When the uncertainty set $\mathcal{P}$ in (7.40) is a finite set of (symmetric positive semidefinite) matrices, *i.e.*,

$$\mathcal{P} = \{P_1, \ldots, P_k\} \subseteq \mathbf{S}_+^n,$$

then minimizing the worst-case objective

$$\sup\{x^T P_i x + q^T x + r \mid i = 1, \ldots, k\}$$

is equivalent to minimizing the maximum of a finite number of quadratic functions. This problem is easily expressed via the epigraph form as

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & (1/2)x^T P_i x + q^T x + r \leq t, \quad i = 1, \ldots, k \\ & Ax \preceq b, \end{array}$$

where $x \in \mathbf{R}^n$ and $t \in \mathbf{R}$ are the variables. Since it is assumed that $P_i \in \mathbf{S}_+^n$ for $i = 1, \ldots, k$, the problem above is a convex optimization problem.

Example 7.13 *Bounded eigenvalues.* Suppose the uncertainty set $\mathcal{P}$ in (7.40) is given by

$$\mathcal{P} = \{P \in \mathbf{S}^n \mid \|P - P_0\|_2 \leq \gamma\},$$

where $P_0 \in \mathbf{S}^n_+$, $\gamma > 0$ are given, and $\|\cdot\|_2$ is the spectral norm of matrices. We can give this uncertainty set a more intuitive interpretation by noticing that the spectral norm of a symmetric matrix is equal to the maximum absolute value of its eigenvalues, *i.e.*,

$$\|P - P_0\|_2 = \sigma_{\max}(P - P_0) = \max_{i=1,\ldots,n} |\lambda_i(P - P_0)|,$$

and hence the uncertainty set $\mathcal{P}$ can be expressed as

$$\mathcal{P} = \{P \in \mathbf{S}^n \mid \max_{i=1,\ldots,n} |\lambda_i(P - P_0)| \leq \gamma\} = \{P \in \mathbf{S}^n \mid -\gamma I \preceq P - P_0 \preceq \gamma I\},$$

which means that the eigenvalues of the difference $P - P_0$ are bounded within the interval $[-\gamma, \gamma]$.

With this uncertainty set, the worst-case quadratic term in the objective can be evaluated as

$$\begin{array}{rcl} \sup_{P\in\mathcal{P}} x^T P x & = & \sup\{x^T P_0 x + x^T (P - P_0)x \mid -\gamma I \preceq P - P_0 \preceq \gamma I\} \\ & = & x^T P_0 x + \sup\{x^T (P - P_0)x \mid -\gamma I \preceq P - P_0 \preceq \gamma I\} \\ & = & x^T P_0 x + \gamma \|x\|_2^2. \end{array}$$

Therefore, the problem (7.40) can be expressed as the following convex quadratic program:

$$\begin{array}{ll} \text{minimize} & (1/2)x^T P_0 x + q^T x + r + \gamma \|x\|_2^2 \\ \text{subject to} & Ax \preceq b, \end{array}$$

where $x \in \mathbf{R}^n$ is the variable. Notice that this problem corresponds to a Tikhonov regularized version of the (convex) quadratic program with quadratic coefficient given by the nominal matrix P_0.

Example 7.14 *Ellipsoid of matrices.* Suppose the uncertainty set $\mathcal{P}$ in (7.40) is an ellipsoid of matrices, *i.e.*,

$$\mathcal{P} = \left\{ P_0 + \sum_{i=1}^k u_i P_i \;\middle|\; \|u\|_2 \leq 1 \right\},$$

where $P_0, P_1, \ldots, P_k \in \mathbf{S}^n_+$ are given (*cf.* example 7.10). Then the worst-case quadratic term in the objective can be evaluated as

$$\begin{array}{rcl} \sup\limits_{P\in\mathcal{P}} x^T P x & = & \sup \left\{ x^T P_0 x + \sum\limits_{i=1}^k u_i (x^T P_i x) \;\middle|\; \|u\|_2 \leq 1 \right\} \\ & = & x^T P_0 x + \sup \left\{ \sum\limits_{i=1}^k u_i (x^T P_i x) \;\middle|\; \|u\|_2 \leq 1 \right\} \\ & = & x^T P_0 x + \left(\sum\limits_{i=1}^k (x^T P_i x)^2 \right)^{1/2}, \end{array}$$

where the last equality follows from the definition of the dual norm. Therefore, the problem (7.40) can be expressed as

$$\begin{array}{ll} \text{minimize} & (1/2)x^T P_0 x + q^T x + r + (1/2)\left(\sum_{i=1}^k \left(x^T P_i x\right)^2\right)^{1/2} \\ \text{subject to} & Ax \preceq b, \end{array} \tag{7.41}$$

where $x \in \mathbf{R}^n$ is the variable.

The problem (7.41) is a convex optimization problem. To see this, we first introduce the epigraph variable $z \in \mathbf{R}^k$ for the last term in the objective, and then the problem (7.41) is equivalent to

$$\begin{array}{ll} \text{minimize} & (1/2)x^T P_0 x + q^T x + r + (1/2)\|z\|_2 \\ \text{subject to} & x^T P_i x \leq z_i, \quad i = 1, \ldots, k \\ & Ax \preceq b, \end{array}$$

where $x \in \mathbf{R}^n$ and $z \in \mathbf{R}^k$ are the variables. Now the objective is convex in (x, z) since it is the sum of a convex quadratic function and a norm function, and the constraints are also convex in (x, z) since $x^T P_i x$ is a convex function of x for each $i = 1, \ldots, k$, so this problem is a convex optimization problem.

The convexity of the problem (7.41) can also be verified by directly checking the convexity of the last term in the objective (since the first term is simply a convex quadratic function). Specifically, let

$$h(x) = (1/2)\left(\sum_{i=1}^k \left(x^T P_i x\right)^2\right)^{1/2},$$

which can be considered as the composition

$$h(x) = g(f_1(x), \ldots, f_k(x)),$$

where the function $g\colon \mathbf{R}^k \to \mathbf{R}$ is defined by $g(z) = (1/2)\|z\|_2$ (with domain $\mathbf{dom}\, g = \mathbf{R}^k_+$) and the functions $f_i\colon \mathbf{R}^n \to \mathbf{R}$ are defined by $f_i(x) = x^T P_i x$ for $i = 1, \ldots, k$. The function g is convex and nondecreasing in each component (on $\mathbf{R}^k_+$), and the functions f_i are convex and nonnegative for each $i = 1, \ldots, k$, so the composition $h(x) = g(f_1(x), \ldots, f_k(x))$ is a convex function of x.

7.2.2 Worst-case robust approximation

The ideas of worst-case robust optimization presented in the previous section apply to linear norm approximation problems where the problem data is uncertain. Here, as an example, we consider the following *worst-case robust approximation problem*:

$$\text{minimize} \quad \sup\{\|Ax - b\|_2 \mid A \in \mathcal{A}\}, \tag{7.42}$$

where $x \in \mathbf{R}^n$ is the variable, $\mathcal{A} \subseteq \mathbf{R}^{m \times n}$ is a given uncertainty set of the feature matrix $A \in \mathbf{R}^{m \times n}$, and $b \in \mathbf{R}^m$ is a given (fixed) response vector. We will discuss several examples where the problem (7.42) can be expressed as a tractable convex optimization problem.

Finite set of feature matrices

When the uncertainty set $\mathcal{A}$ in (7.42) is a finite set of matrices, *i.e.*,

$$\mathcal{A} = \{A_1, \ldots, A_k\} \subseteq \mathbf{R}^{m\times n}, \tag{7.43}$$

the problem (7.42) reduces to

$$\text{minimize} \quad \max_{i=1,\ldots,k} \|A_i x - b\|_2, \tag{7.44}$$

which can be expressed in the epigraph form as

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \|A_i x - b\|_2 \leq t, \quad i = 1, \ldots, k, \end{array}$$

where $x \in \mathbf{R}^n$ and $t \in \mathbf{R}$ are the variables. In fact, we can show that the problem (7.44) also applies to the case where the uncertainty set $\mathcal{A}$ given by (7.43) is replaced by its convex hull, *i.e.*,

$$\mathcal{A} = \mathbf{conv}\{A_1, \ldots, A_k\} = \left\{ \sum_{i=1}^k \theta_i A_i \;\middle|\; \theta \in \mathbf{R}^k,\ \theta \succeq 0,\ \mathbf{1}^T\theta = 1 \right\}.$$

(See exercise 7.3.)

Ellipsoids of feature vectors

Suppose the uncertainty set $\mathcal{A}$ in (7.42) is given by

$$\mathcal{A} = \left\{ A = \begin{bmatrix} a_1 & \cdots & a_n \end{bmatrix}^T \in \mathbf{R}^{m\times n} \;\middle|\; a_i \in \mathcal{E}_i,\ i = 1, \ldots, m \right\}, \tag{7.45}$$

where

$$\mathcal{E}_i = \{\bar{a}_i + P_i u \mid \|u\|_2 \leq 1\} \subseteq \mathbf{R}^n, \quad i = 1, \ldots, m,$$

and the matrices $P_i \in \mathbf{R}^{n\times n}$ and the vectors $\bar{a}_i \in \mathbf{R}^n$ for $i = 1, \ldots, m$ are given. This uncertainty structure means that the ith feature vector (*i.e.*, the ith row of the feature matrix A) is known to lie within some ellipsoid $\mathcal{E}_i$ in $\mathbf{R}^n$ for each $i = 1, \ldots, m$. Recall (from example 7.10) that when P_i is singular, the ellipsoid $\mathcal{E}_i$ is flat with dimension $\mathbf{rank}\, P_i$, and when $P_i = 0$, there is no uncertainty in the ith feature vector.

With this uncertainty set, the objective of (7.42) can be evaluated as

$$\sup_{A\in\mathcal{A}} \|Ax - b\|_2 = \sup_{A\in\mathcal{A}} \left(\sum_{i=1}^m (a_i^T x - b_i)^2 \right)^{1/2} = \left(\sum_{i=1}^m \sup_{a_i\in\mathcal{E}_i} (a_i^T x - b_i)^2 \right)^{1/2}.$$

Noticing that

$$\sup_{a_i\in\mathcal{E}_i} (a_i^T x - b_i)^2 = \left(\sup_{a_i\in\mathcal{E}_i} |a_i^T x - b_i| \right)^2,$$

evaluating the objective of the problem (7.42) reduces to evaluating the supremum of the absolute value of the linear function $a_i^T x - b_i$ over $a_i \in \mathcal{E}_i$, for all $i = 1, \ldots, m$. Since

$$\begin{aligned}
\sup_{a_i \in \mathcal{E}_i} |a_i^T x - b_i| &= \sup\{|(\bar{a}_i + P_i u)^T x - b_i| \mid \|u\|_2 \leq 1\} \\
&= \sup\{|\bar{a}_i^T x - b_i + u^T P_i^T x| \mid \|u\|_2 \leq 1\} \\
&= |\bar{a}_i^T x - b_i| + \sup\{|u^T P_i^T x| \mid \|u\|_2 \leq 1\} \\
&= |\bar{a}_i^T x - b_i| + \|P_i^T x\|_2,
\end{aligned}$$

where the last equality follows from the definition of the dual norm (see remark A.2), we have

$$\sup_{A \in \mathcal{A}} \|Ax - b\|_2 = \left(\sum_{i=1}^m \left(|\bar{a}_i^T x - b_i| + \|P_i^T x\|_2 \right)^2 \right)^{1/2}.$$

As a result, the problem (7.42) under the uncertainty set $\mathcal{A}$ given by (7.45) can be expressed as

$$\begin{array}{ll}
\text{minimize} & \|t\|_2 \\
\text{subject to} & |\bar{a}_i^T x - b_i| + \|P_i^T x\|_2 \leq t_i, \quad i = 1, \ldots, m,
\end{array}$$

or equivalently,

$$\begin{array}{ll}
\text{minimize} & \|t\|_2 \\
\text{subject to} & \bar{a}_i^T x - b_i + \|P_i^T x\|_2 \leq t_i, \quad i = 1, \ldots, m \\
& -\bar{a}_i^T x + b_i + \|P_i^T x\|_2 \leq t_i, \quad i = 1, \ldots, m,
\end{array}$$

where $x \in \mathbf{R}^n$ and $t \in \mathbf{R}^m$ are the variables.

Ellipsoid of feature matrices

Suppose the uncertainty set $\mathcal{A}$ in (7.42) is an ellipsoid of feature matrices, *i.e.*,

$$\mathcal{A} = \left\{ A_0 + \sum_{i=1}^k u_i A_i \;\middle|\; \|u\|_2 \leq 1 \right\}, \tag{7.46}$$

where $A_0, A_1, \ldots, A_k \in \mathbf{R}^{m \times n}$ are given. We consider a simple case where $k = 1$, *i.e.*, $u \in \mathbf{R}$ is scalar, so the set (7.46) reduces to

$$\mathcal{A} = \{A_0 + uA_1 \mid |u| \leq 1\}.$$

This uncertainty structure essentially means that the feature matrix A is known to lie within the line segment connecting the two matrices $A_0 - A_1$ and $A_0 + A_1$ in $\mathbf{R}^{m \times n}$. Therefore, the worst-case objective in the problem (7.42) is given by

$$\begin{aligned}
\sup_{A \in \mathcal{A}} \|Ax - b\|_2 &= \sup\{\|(A_0 + uA_1)x - b\|_2 \mid |u| \leq 1\} \\
&= \max\{\|(A_0 - A_1)x - b\|_2, \|(A_0 + A_1)x - b\|_2\},
\end{aligned}$$

where the last equality follows from the fact that $\|(A_0 + uA_1)x - b\|_2$ is a convex function of u, and hence its maximum over the interval $[-1, 1]$ is attained at one of the endpoints $u = -1$ and $u = 1$. The problem (7.42) corresponding to this objective can then be expressed as

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \|(A_0 - A_1)x - b\|_2 \leq t \\ & \|(A_0 + A_1)x - b\|_2 \leq t, \end{array}$$

where $x \in \mathbf{R}^n$ and $t \in \mathbf{R}$ are the variables.

When the uncertainty set $\mathcal{A}$ is given by (7.46) with $k > 1$, the problem (7.42) is still tractable, but this result is not easy to show; see the references.

Numerical example

We consider a numerical example to compare the ideas of stochastic, worst-case robust, and nominal least squares in the context of linear approximation. In particular, consider the system of linear equations

$$A(u)x = b \tag{7.47}$$

in the variable $x \in \mathbf{R}^n$, where the data matrix $A(u) \in \mathbf{R}^{m\times n}$ is parameterized by $u \in \mathbf{R}$ as

$$A(u) = A_0 + uA_1, \quad u \in [-1, 1], \tag{7.48}$$

and $A_0, A_1 \in \mathbf{R}^{m\times n}$ are given. The vector $b \in \mathbf{R}^m$ is given and fixed.

The nominal least squares applied to the system (7.47) simply ignores the variation of the data matrix $A(u)$ and seeks for a solution that minimizes the approximation error with respect to the nominal data matrix A_0, which corresponds to the problem

$$\text{minimize} \quad \|A_0x - b\|_2^2, \tag{7.49}$$

where $x \in \mathbf{R}^n$ is the variable. Assuming $m \geq n$ and A_0 has full rank, the solution to this problem is given by

$$x^{\text{nom}} = (A_0^T A_0)^{-1} A_0^T b.$$

In stochastic least squares, we assume that the parameter u in (7.48) is a uniformly distributed random variable over the interval $[-1, 1]$, and we seek for a solution that minimizes the expected approximation error, which corresponds to the problem

$$\text{minimize} \quad \mathbf{E}\,\|A(u)x - b\|_2^2, \tag{7.50}$$

where $x \in \mathbf{R}^n$ is the variable. Recall from example 7.1 that this stochastic program is equivalent to

$$\text{minimize} \quad \|A_0x - b\|_2^2 + \|P^{1/2}x\|_2^2,$$

where

$$P = \mathbf{E}\,(uA_1)^T(uA_1) = \mathbf{E}\,u^2 A_1^T A_1 = (1/3)A_1^T A_1.$$

Assuming $A_0^T A_0 + P$ is invertible, the solution to this problem is given by

$$x^{\text{sto}} = \left(A_0^T A_0 + P\right)^{-1} A_0^T b.$$

Finally, the in worst-case robust least squares, we seek for a solution that minimizes the worst-case approximation error, which corresponds to the problem

$$\text{minimize} \quad \sup_{u\in[-1,1]} \|A(u)x - b\|_2^2, \tag{7.51}$$

where $x \in \mathbf{R}^n$ is the variable. According to the discussions in the previous section, this problem can be expressed as

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \|(A_0 - A_1)x - b\|_2^2 \leq t \\ & \|(A_0 + A_1)x - b\|_2^2 \leq t, \end{array}$$

where $x \in \mathbf{R}^n$ and $t \in \mathbf{R}$ are the variables. We denote the solution to this problem by x^{wc}.

We solve the problems (7.49), (7.50), and (7.51) for a randomly generated instance with $m = 25$ and $n = 20$, where $\|A_0\|_2 = 10$ and $\|A_1\|_2 = 1$ (so the variation in the data matrix $A(u)$ is roughly 10% of A_0). Figure 7.6 shows the approximation errors $\|A(u)x - b\|_2$ of the three solutions x^{nom}, x^{sto}, and x^{wc} as functions of the parameter u, which illustrates the sensitivity of the approximation error to the variation in the data matrix $A(u)$ for the three solutions. Among the three solutions, the nominal least squares solution x^{nom} has the smallest approximation error at the nominal data matrix A_0 (corresponding to $u = 0$), but its approximation error is very sensitive to perturbations in the data matrix, *i.e.*, the error increases significantly as u deviates from 0 and approaches ± 1. The worst-case robust least squares solution x^{wc} has the largest approximation error at the nominal data matrix A_0, but it leads to the smallest approximation error at the endpoints $u = \pm 1$ (corresponding to the worst-case scenarios), and the error does not increase much as u varies within $[-1, 1]$. The stochastic least squares solution x^{sto} balances between the nominal performance and the worst-case performance, and it has the smallest expected approximation error across $u \in [-1, 1]$.

Figure 7.7 shows the distributions of the approximation errors $\|A(u)x - b\|_2$ for the three solutions x^{nom}, x^{sto}, and x^{wc} when u is uniformly distributed over $[-1, 1]$. Notice that the approximation error amplitudes corresponding to the nominal least squares solution x^{nom} spread most widely, while the approximation error amplitudes corresponding to the stochastic and worst-case robust least squares solutions x^{sto} and x^{wc} are more concentrated around a smaller value, which again illustrates that the stochastic and worst-case robust least squares solutions are less sensitive to the variation in the data matrix $A(u)$ than the nominal least squares solution.

7.3 Robust linear discrimination

We consider the problem of linear discrimination introduced in §4.4. Suppose we are given a dataset consisting of two groups of points $\{x_1, \ldots, x_M\}$ and $\{y_1, \ldots, y_N\}$ in

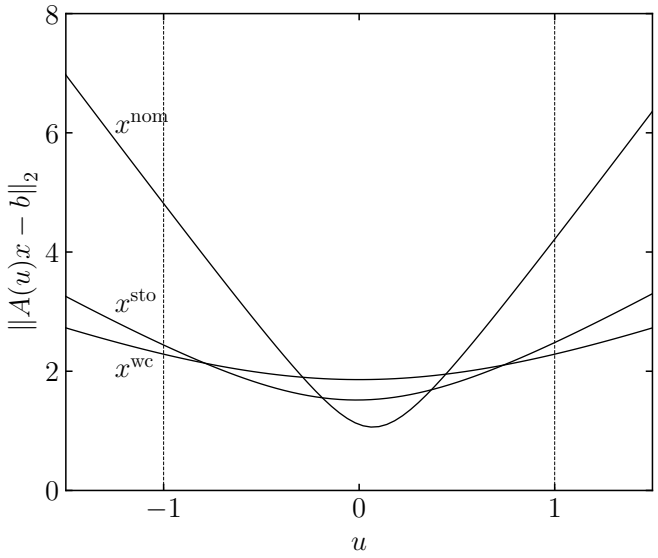

Figure 7.6 Approximation errors $\|A(u)x-b\|_2$ of the nominal, stochastic, and worst-case robust least squares solution x^{nom}, x^{sto}, and x^{wc} (from the problems (7.49), (7.50), and (7.51), respectively) as functions of the parameter $u \in \mathbf{R}$ in the data matrix $A(u) = A_0 + uA_1$. The dashed vertical lines indicate the interval $[-1, 1]$ of the parameter u, in which the data matrix $A(u)$ is assumed to vary.

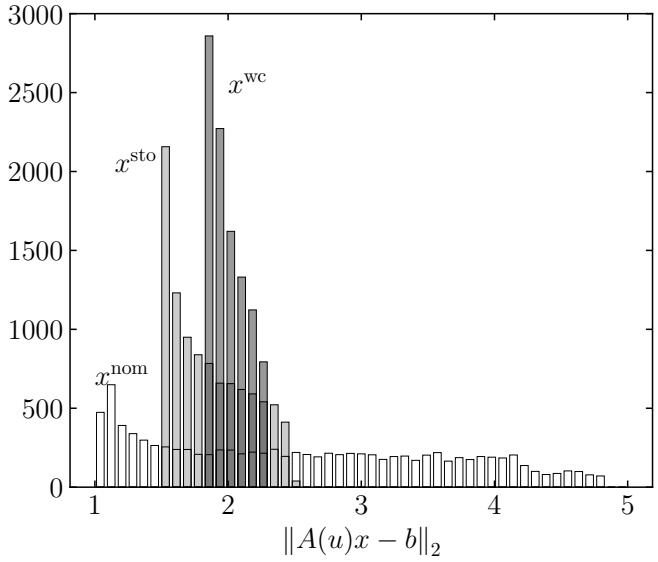

Figure 7.7 Distributions of the approximation errors $\|A(u)x-b\|_2$ under the same x^{nom}, x^{sto}, and x^{wc} as in figure 7.6, when the parameter u in the data matrix $A(u) = A_0 + uA_1$ is uniformly distributed over the interval $[-1, 1]$.

$\mathbf{R}^n$. A *linear discrimination* problem aims at finding an affine function $f: \mathbf{R}^n \to \mathbf{R}$ of the form $f(x) = a^T x - b$ such that

$$f(x_i) > 0, \quad i = 1, \ldots, M,$$

and

$$f(y_i) < 0, \quad i = 1, \ldots, N.$$

This can be formulated as the feasibility problem

$$\begin{array}{ll} \text{find} & (a, b) \\ \text{subject to} & a^T x_i - b > 0, \quad i = 1, \ldots, M \\ & a^T y_i - b < 0, \quad i = 1, \ldots, N, \end{array} \tag{7.52}$$

where $a \in \mathbf{R}^n$ and $b \in \mathbf{R}$ are the optimization variables. The basic feasibility problem (7.52) usually has infinitely many solutions. Hence, it is reasonable to seek for a solution that is in some sense *robust*.

Remark 7.2 *Solution set of linear discrimination.* The solution set of the problem (7.52) is a convex polyhedral cone in $\mathbf{R}^{n+1}$. To show this, let

$$z = \begin{bmatrix} a \\ b \end{bmatrix}, \quad \tilde{x}_i = \begin{bmatrix} x_i \\ -1 \end{bmatrix}, \quad \tilde{y}_i = \begin{bmatrix} y_i \\ -1 \end{bmatrix}.$$

Then the solution set of the problem (7.52) can be expressed as

$$\left\{ z \in \mathbf{R}^{n+1} \;\middle|\; \begin{array}{ll} z^T \tilde{x}_i > 0, & i = 1, \ldots, M \\ z^T \tilde{y}_i < 0, & i = 1, \ldots, N \end{array} \right\},$$

which is the intersection of $M + N$ halfspaces in $\mathbf{R}^{n+1}$, and hence is a polyhedron. The solution set is also a cone (without the origin) since if z is a solution, then αz is also a solution for any $\alpha > 0$.

Noticing that the problem (7.52) is equivalent to the problem

$$\begin{array}{ll} \text{find} & (a, b) \\ \text{subject to} & a^T x_i - b \geq 1, \quad i = 1, \ldots, M \\ & a^T y_i - b \leq -1, \quad i = 1, \ldots, N \end{array} \tag{7.53}$$

(see exercise 4.7), and since the problem (7.53) is homogeneous in the variables a and b, it is natural to seek for a solution that minimizes the norm of the normal vector a of the separating hyperplane. This leads to the following optimization problem:

$$\begin{array}{ll} \text{minimize} & \|a\|_2 \\ \text{subject to} & a^T x_i - b \geq 1, \quad i = 1, \ldots, M \\ & a^T y_i - b \leq -1, \quad i = 1, \ldots, N \end{array} \tag{7.54}$$

with variables $a \in \mathbf{R}^n$ and $b \in \mathbf{R}$, which is a convex program with convex objective and linear inequality constraints.

7.3.1 Geometric interpretation

In fact, the problem (7.54) has a very nice geometric interpretation. To see this, we first perform a change of variable $a = \tilde{a}/t$ and $b = \tilde{b}/t$, where $t > 0$ is a scaling factor. Then the problem (7.54) can be rewritten as

$$\begin{array}{ll} \text{minimize} & \|\tilde{a}\|_2/t \\ \text{subject to} & \tilde{a}^T x_i - \tilde{b} \geq t, \quad i = 1, \ldots, M \\ & \tilde{a}^T y_i - \tilde{b} \leq -t, \quad i = 1, \ldots, N \\ & t > 0 \end{array} \tag{7.55}$$

with variables $\tilde{a} \in \mathbf{R}^n$, $\tilde{b} \in \mathbf{R}$, and $t \in \mathbf{R}$. Notice that for any feasible point $(\tilde{a}, \tilde{b}, t)$, if we scale all the variables by some positive factor $\alpha > 0$, then the resulting point $(\alpha\tilde{a}, \alpha\tilde{b}, \alpha t)$ is also a feasible point with exactly the same objective value. Hence, the problem (7.55) also has infinitely many solutions. To remove this ambiguity, we can further impose the constraint $\|\tilde{a}\|_2 \leq 1$, then the problem (7.55) becomes

$$\begin{array}{ll} \text{maximize} & t \\ \text{subject to} & \tilde{a}^T x_i - \tilde{b} \geq t, \quad i = 1, \ldots, M \\ & \tilde{a}^T y_i - \tilde{b} \leq -t, \quad i = 1, \ldots, N \\ & t > 0, \quad \|\tilde{a}\|_2 \leq 1. \end{array} \tag{7.56}$$

This problem is also a convex optimization problem, with linear objective, $M + N + 1$ linear inequality constraints, and one convex quadratic inequality constraint. Noticing that since the first two inequality constraints in the problem (7.56) are homogeneous in the problem variables, the last inequality constraint $\|\tilde{a}\|_2 \leq 1$ must be tight at the optimal point, *i.e.*, we have $\|\tilde{a}^\star\|_2 = 1$ when $(\tilde{a}^\star, \tilde{b}^\star, t^\star)$ is an optimal point of the problem (7.56).

Remark 7.3 To prove the equivalence between the problem (7.56) and the original problem (7.54), suppose $(\tilde{a}^\star, \tilde{b}^\star, t^\star)$ is an optimal point of the problem (7.56), and suppose $(a^\star, b^\star)$ is an optimal point of the problem (7.54). Let $a = \tilde{a}^\star/t^\star$ and $b = \tilde{b}^\star/t^\star$, then (a, b) must be a feasible point of the problem (7.54) with objective value

$$\|a\|_2 = \|\tilde{a}^\star\|_2/t^\star = 1/t^\star.$$

Since $(a^\star, b^\star)$ is an optimal point of the problem (7.54), we have

$$\|a^\star\|_2 \leq \|a\|_2 = 1/t^\star. \tag{7.57}$$

Conversely, let $\tilde{a} = a^\star/\|a^\star\|_2$, $\tilde{b} = b^\star/\|a^\star\|_2$, and $t = 1/\|a^\star\|_2$, then $(\tilde{a}, \tilde{b}, t)$ must be a feasible point of the problem (7.56) with objective value

$$t = 1/\|a^\star\|_2.$$

Similarly, since $(\tilde{a}^\star, \tilde{b}^\star, t^\star)$ is an optimal point of the problem (7.56), we have

$$t^\star \geq t = 1/\|a^\star\|_2,$$

i.e.,

$$\|a^\star\|_2 \geq 1/t^\star. \tag{7.58}$$

Put together the inequalities (7.57) and (7.58), we have

$$\|a^\star\|_2 = 1/t^\star.$$

This says that, the optimal point $(a^\star, b^\star)$ of the problem (7.54) can be obtained from the optimal point $(\tilde{a}^\star, \tilde{b}^\star, t^\star)$ of the problem (7.56) by performing the change of variable $a^\star = \tilde{a}^\star/t^\star$ and $b^\star = \tilde{b}^\star/t^\star$, and vice versa (with $t^\star = 1/\|a^\star\|_2$).

We can give the problem (7.56) the following geometric interpretation. Let $(\tilde{a}, \tilde{b}, t)$ be a feasible point of the problem (7.56). If $\|\tilde{a}\|_2 = 1$ (which is necessarily true at the optimal point), then $\tilde{a}^T x_i - \tilde{b}$ is the distance from the point x_i to the hyperplane

$$\{z \in \mathbf{R}^n \mid \tilde{a}^T z = \tilde{b}\},$$

and $\tilde{b} - \tilde{a}^T y_i$ is the distance from the point y_i to the same hyperplane. Therefore, the problem (7.56) can be interpreted as finding a hyperplane that separates the two groups of points $\{x_1, \ldots, x_M\}$ and $\{y_1, \ldots, y_N\}$ that has the largest distance to the closest point in either group, *i.e.*, finding a separating hyperplane with the largest *margin* between the groups. In other words, the problem (7.56) finds the thickest *slab* (*i.e.*, the region between two parallel hyperplanes) that separates the two groups of points, where the optimal value $t^\star$ of the problem (7.56) corresponds to the half width of this slab (see also exercise 7.5). Because of this geometric interpretation, the problem (7.56) is sometimes called the *maximum margin linear discrimination* problem.

Figure 7.8 illustrates this geometric interpretation of the problem (7.56) on a randomly generated dataset in $\mathbf{R}^2$.

Relaxing the separability constraint

In practice, the strict constraint $t > 0$ in the problem (7.56) is usually relaxed to the nonnegative constraint $t \geq 0$, or simply removed so that $t \in \mathbf{R}$ (which we will see soon that are equivalent). In the second case, the problem (7.56) becomes

$$\begin{array}{ll} \text{maximize} & t \\ \text{subject to} & \tilde{a}^T x_i - \tilde{b} \geq t, \quad i = 1, \ldots, M \\ & \tilde{a}^T y_i - \tilde{b} \leq -t, \quad i = 1, \ldots, N \\ & \|\tilde{a}\|_2 \leq 1, \end{array} \tag{7.59}$$

where $\tilde{a} \in \mathbf{R}^n$, $\tilde{b} \in \mathbf{R}$, and $t \in \mathbf{R}$ are the optimization variables.

We can show that the optimal value $t^\star$ of the relaxed problem (7.59) is positive if and only if the original problem (7.56) is feasible, *i.e.*, if and only if the two groups of points $\{x_1, \ldots, x_M\}$ and $\{y_1, \ldots, y_N\}$ are linearly separable. Let $(\tilde{a}^\star, \tilde{b}^\star, t^\star)$ be an optimal point of the problem (7.59). If $t^\star > 0$, then we have

$$\tilde{a}^{\star T} x_i \geq t^\star + \tilde{b}^\star > \tilde{b}^\star > \tilde{b}^\star - t^\star \geq \tilde{a}^{\star T} y_j$$

for all $i = 1, \ldots, M$ and $j = 1, \ldots, N$, which means that $\{z \mid \tilde{a}^{\star T} z = \tilde{b}^\star\}$ defines a separating hyperplane of the two groups of points $\{x_1, \ldots, x_M\}$ and $\{y_1, \ldots, y_N\}$.

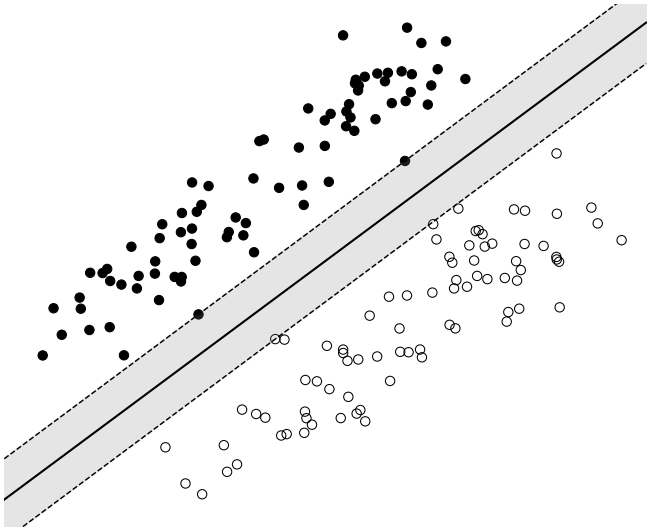

Figure 7.8 The dots and circles represent two groups of points in $\mathbf{R}^2$ that are linearly separable. The solid line represents the separating hyperplane (which is a line in $\mathbf{R}^2$) that solves the problem (7.56), and the two dashed lines represent the two parallel hyperplanes that define the thickest slab (shown shaded) that separates the two groups of points. The distance from the solid line to either of the dashed lines is the optimal value $t^\star$ of the problem (7.56).

Conversely, if the two groups of points are linearly separable, then the problem (7.56) is feasible, and hence there is a positive t satisfying the constraints of the problem (7.59), so the optimal value $t^\star$ of the problem (7.59) must be positive.

According to the previous discussion, the problems (7.56) and (7.59) are equivalent when the given two groups of points $\{x_1, \ldots, x_M\}$ and $\{y_1, \ldots, y_N\}$ are linearly separable. On the other hand, if $t^\star = 0$ achieves the optimal value of the relaxed problem (7.59), then the original problem (7.56) must be infeasible, where in this case, the two groups of points $\{x_1, \ldots, x_M\}$ and $\{y_1, \ldots, y_N\}$ are not linearly separable. Note that the optimal value $t^\star$ of the relaxed problem (7.59) cannot be negative, since the point $(\tilde{a}, \tilde{b}, t) = (0, 0, 0)$ is always feasible.

The separating hyperplane shown in figure 7.8 was obtained by solving the relaxed problem (7.59) as a surrogate to the original problem (7.56).

7.3.2 Interpretation via Lagrange duality

Intuitively (and also indicated by figure 7.8), when the given two groups of points $\{x_1, \ldots, x_M\}$ and $\{y_1, \ldots, y_N\}$ are linearly separable, the optimal value $t^\star$ of the problem (7.59) corresponds to the half distance between the convex hulls of the two groups of points $\{x_1, \ldots, x_M\}$ and $\{y_1, \ldots, y_N\}$, *i.e.*,

$$\begin{aligned} t^\star &= (1/2)\,\mathbf{dist}(\mathbf{conv}\{x_1, \ldots, x_M\},\ \mathbf{conv}\{y_1, \ldots, y_N\}) \\ &= \frac{1}{2}\inf\left\{ \|x - y\|_2 \;\middle|\; \begin{array}{l} x \in \mathbf{conv}\{x_1, \ldots, x_M\} \\ y \in \mathbf{conv}\{y_1, \ldots, y_N\} \end{array} \right\}. \end{aligned} \tag{7.60}$$

This result can be shown by analyzing the dual problem of the problem (7.59).

The Lagrangian (for the problem of minimizing $-t$) of the problem (7.59) is expressed as

$$\begin{aligned} & L((\tilde{a}, \tilde{b}, t), (u, v, w)) \\ = \;& -t + \sum_{i=1}^{M} u_i(t - \tilde{a}^T x_i + \tilde{b}) + \sum_{i=1}^{N} v_i(t + \tilde{a}^T y_i - \tilde{b}) + w(\|\tilde{a}\|_2 - 1) \\ = \;& -t + (t + \tilde{b})\mathbf{1}^T u + (t - \tilde{b})\mathbf{1}^T v - \tilde{a}^T \left(\sum_{i=1}^{M} u_i x_i - \sum_{i=1}^{N} v_i y_i \right) + w(\|\tilde{a}\|_2 - 1) \\ = \;& (\mathbf{1}^T u + \mathbf{1}^T v - 1)t + (\mathbf{1}^T u - \mathbf{1}^T v)\tilde{b} - \tilde{a}^T \left(\sum_{i=1}^{M} u_i x_i - \sum_{i=1}^{N} v_i y_i \right) + w(\|\tilde{a}\|_2 - 1), \end{aligned}$$

where $u \in \mathbf{R}^M$, $v \in \mathbf{R}^N$, and $w \in \mathbf{R}$ are the dual variables.

To obtain the dual function, we need to minimize the Lagrangian over the primal variables $(\tilde{a}, \tilde{b}, t)$. Noticing that the Lagrangian is linear in t and $\tilde{b}$, for any dual feasible point (u, v, w), we must have

$$(\mathbf{1}^T u + \mathbf{1}^T v - 1)t + (\mathbf{1}^T u - \mathbf{1}^T v)\tilde{b} = 0,$$

i.e.,

$$\mathbf{1}^T u + \mathbf{1}^T v = 1, \quad \mathbf{1}^T u = \mathbf{1}^T v, \tag{7.61}$$

since otherwise the Lagrangian can be made arbitrarily negative by scaling t and $\tilde{b}$, and the dual function would hence be unbounded below. The condition (7.61) therefore implies the constraints

$$\mathbf{1}^T u = 1/2, \qquad \mathbf{1}^T v = 1/2, \tag{7.62}$$

on the dual variables u and v.

When (7.62) holds, the dual function g of the problem (7.59) is expressed as

$$\begin{aligned} g(u,v,w) &= \inf_{(\tilde{a},\tilde{b},t)} L((\tilde{a},\tilde{b},t),(u,v,w)) \\ &= \inf_{\tilde{a}} \left(-\tilde{a}^T \left(\sum_{i=1}^M u_i x_i - \sum_{i=1}^N v_i y_i \right) + w(\|\tilde{a}\|_2 - 1) \right) \\ &= -w + \inf_{\tilde{a}} \left(-\tilde{a}^T \left(\sum_{i=1}^M u_i x_i - \sum_{i=1}^N v_i y_i \right) + w\|\tilde{a}\|_2 \right). \end{aligned}$$

To evaluate the infimum over $\tilde{a}$, we can use the Cauchy-Schwarz inequality to obtain

$$\tilde{a}^T \left(\sum_{i=1}^M u_i x_i - \sum_{i=1}^N v_i y_i \right) \leq \|\tilde{a}\|_2 \left\| \sum_{i=1}^M u_i x_i - \sum_{i=1}^N v_i y_i \right\|_2,$$

and hence

$$\begin{aligned} -\tilde{a}^T \left(\sum_{i=1}^M u_i x_i - \sum_{i=1}^N v_i y_i \right) + w\|\tilde{a}\|_2 &\geq -\|\tilde{a}\|_2 \left\| \sum_{i=1}^M u_i x_i - \sum_{i=1}^N v_i y_i \right\|_2 + w\|\tilde{a}\|_2 \\ &= \|\tilde{a}\|_2 \left(w - \left\| \sum_{i=1}^M u_i x_i - \sum_{i=1}^N v_i y_i \right\|_2 \right). \end{aligned}$$

When $w \geq \left\| \sum_{i=1}^M u_i x_i - \sum_{i=1}^N v_i y_i \right\|_2$, the right-hand side of the above inequality is nonnegative for any $\tilde{a} \in \mathbf{R}^n$, so the infimum over $\tilde{a}$ is 0 (which is achieved at $\tilde{a} = 0$). On the other hand, when $w < \left\| \sum_{i=1}^M u_i x_i - \sum_{i=1}^N v_i y_i \right\|_2$, by taking $\tilde{a} = \lambda \left(\sum_{i=1}^M u_i x_i - \sum_{i=1}^N v_i y_i \right)$ for some $\lambda > 0$, the above inequality is tight and the right-hand side goes to $-\infty$ as $\lambda \to \infty$, so in this case, the infimum over $\tilde{a}$ is $-\infty$. Therefore, the dual function g reduces to

$$g(u,v,w) = \begin{cases} -w, & w \geq \left\| \sum_{i=1}^M u_i x_i - \sum_{i=1}^N v_i y_i \right\|_2 \\ -\infty, & \text{otherwise}, \end{cases}$$

so the dual problem of the problem (7.59) is

$$\begin{array}{ll} \text{maximize} & -w \\ \text{subject to} & w \geq \left\| \sum_{i=1}^M u_i x_i - \sum_{i=1}^N v_i y_i \right\|_2 \\ & \mathbf{1}^T u = 1/2, \quad \mathbf{1}^T v = 1/2 \\ & u \succeq 0, \quad v \succeq 0 \end{array}$$

with variables $u \in \mathbf{R}^M$, $v \in \mathbf{R}^N$, and $w \in \mathbf{R}$. This is essentially the epigraph form of the problem

$$\begin{array}{ll} \text{minimize} & \left\| \sum_{i=1}^M u_i x_i - \sum_{i=1}^N v_i y_i \right\|_2 \\ \text{subject to} & \mathbf{1}^T u = 1/2, \quad \mathbf{1}^T v = 1/2 \\ & u \succeq 0, \quad v \succeq 0 \end{array} \tag{7.63}$$

with variables $u \in \mathbf{R}^M$ and $v \in \mathbf{R}^N$.

The constraints in the dual problem (7.63) imply that $2\sum_{i=1}^M u_i x_i$ is a convex combination of the points $\{x_1, \ldots, x_M\}$, and $2\sum_{i=1}^N v_i y_i$ is a convex combination of the points $\{y_1, \ldots, y_N\}$, so the objective of the dual problem is to minimize the half distance between a point in the convex hull of $\{x_1, \ldots, x_M\}$ and a point in the convex hull of $\{y_1, \ldots, y_N\}$, *i.e.*,

$$\begin{array}{ll} \text{minimize} & (1/2)\|x - y\|_2 \\ \text{subject to} & x \in \mathbf{conv}\{x_1, \ldots, x_M\} \\ & y \in \mathbf{conv}\{y_1, \ldots, y_N\}. \end{array} \tag{7.64}$$

Given the fact that strong duality holds for the problem (7.59), the optimal value $t^\star$ of the primal problem (7.59) is equivalent to the optimal value of the dual problem (7.64), from which (7.60) follows.

7.3.3 Robustness to weight perturbations

Given two groups of points $\{x_1, \ldots, x_M\}$ and $\{y_1, \ldots, y_N\}$ in $\mathbf{R}^n$ that are linearly separable, and let $f(x) = a^T x - b$ be an affine function that separates the two groups of points. The *weight error margin* of this separating hyperplane is defined as the norm of the smallest $w \in \mathbf{R}^n$ such that the affine function

$$\tilde{f}(x) = (a + w)^T x - b \tag{7.65}$$

fails to separate the two groups of points. In this context, the vector a is often called the *weight vector*. An alternative robustness criterion in robust linear discrimination involves finding a separating hyperplane that maximizes the weight error margin, *i.e.*, finding the separating hyperplane that is most robust to perturbations in the weight vector a.

To formulate this problem, for fixed separating hyperplane parameters a and b, let $t > 0$ be the weight error margin, then according to the definition, we have the separation conditions

$$(a + w)^T x_i - b \geq 0, \quad i = 1, \ldots, M, \tag{7.66}$$

and

$$(a+w)^T y_i - b \leq 0, \quad i = 1, \ldots, N, \tag{7.67}$$

hold for all $w \in \mathbf{R}^n$ with $\|w\|_2 \leq t$. Hence, according to the Cauchy-Schwarz inequality, we have

$$w^T x_i \geq -\|w\|_2 \|x_i\|_2 \geq -t\|x_i\|_2, \quad i = 1, \ldots, M,$$

and

$$w^T y_i \leq \|w\|_2 \|y_i\|_2 \leq t\|y_i\|_2, \quad i = 1, \ldots, N.$$

This shows that the separation conditions (7.66) and (7.67) are implied by the following inequalities:

$$a^T x_i - t\|x_i\|_2 \geq b, \quad i = 1, \ldots, M,$$

and

$$a^T y_i + t\|y_i\|_2 \leq b, \quad i = 1, \ldots, N.$$

Therefore, the *maximum weight error margin linear discrimination* problem can be formulated as

$$\begin{array}{ll} \text{maximize} & t \\ \text{subject to} & a^T x_i - b \geq t\|x_i\|_2, \quad i = 1, \ldots, M \\ & a^T y_i - b \leq -t\|y_i\|_2, \quad i = 1, \ldots, N \\ & \|a\|_2 \leq 1 \end{array} \tag{7.68}$$

with variables $a \in \mathbf{R}^n$, $b \in \mathbf{R}$, and $t \in \mathbf{R}$. Note that, again, the last inequality constraint $\|a\|_2 \leq 1$ is necessary to remove the ambiguity in scaling the variables a, b, and t. The problem (7.68) is a convex optimization problem, with linear objective, $M+N$ linear inequality constraints, and one convex quadratic inequality constraint.

The problem (7.68) can be written in the form of (7.54); see exercise 7.6. We can also show (*cf.* page 275) that the optimal value $t^\star$ of the problem (7.68) is positive if and only if the two groups of points $\{x_1, \ldots, x_M\}$ and $\{y_1, \ldots, y_N\}$ are linearly separable.

Remark 7.4 *Geometric interpretation.* We consider a special case of the problem (7.68) to provide a geometric interpretation. Suppose the given two groups of points $\{x_1, \ldots, x_M\}$ and $\{y_1, \ldots, y_N\}$ are separable by the linear function $f(x) = a^T x$ (which happens, *e.g.*, when the two groups of points are centered at the origin), and consider the following special case of the problem (7.68):

$$\begin{array}{ll} \text{maximize} & t \\ \text{subject to} & a^T x_i \geq t\|x_i\|_2, \quad i = 1, \ldots, M \\ & a^T y_i \leq -t\|y_i\|_2, \quad i = 1, \ldots, N \\ & \|a\|_2 \leq 1 \end{array} \tag{7.69}$$

with variables $a \in \mathbf{R}^n$ and $t \in \mathbf{R}$. Let (a, t) be a feasible point of the problem (7.69) with $\|a\|_2 = 1$ (which is necessarily true at the optimal point), then we can define

$$\cos\alpha_i = a^T x_i / \|x_i\|_2, \quad i = 1, \ldots, M,$$

which is the cosine of the angle between the point x_i and the normal vector a of the separating hyperplane $\{z \in \mathbf{R}^n \mid a^T z = 0\}$, and

$$\cos\beta_i = -a^T y_i / \|y_i\|_2, \quad i = 1, \ldots, N,$$

which is the cosine of the angle between the point y_i and the normal vector $-a$ of the same separating hyperplane. In both cases, this cosine value is nonnegative and is equivalent to the *sine* of the angle θ_i between the corresponding data point $v_i \in \{x_1, \ldots, x_M\} \cup \{y_1, \ldots, y_N\}$ and the separating hyperplane itself, *i.e.*,

$$\sin\theta_i = \begin{cases} \cos\alpha_i = a^T v_i / \|v_i\|_2, & v_i \in \{x_1, \ldots, x_M\} \\ \cos\beta_i = -a^T v_i / \|v_i\|_2, & v_i \in \{y_1, \ldots, y_N\}. \end{cases}$$

This idea is illustrated in figure 7.9. Therefore, we can give the problem (7.69) the following geometric interpretation: Unlike the problem (7.59) which tries to maximize the *distance* between the separating hyperplane and the two groups of points $\{x_1, \ldots, x_M\}$ and $\{y_1, \ldots, y_N\}$, the problem (7.69) tries to find a hyperplane that separates the data points that leads to the largest worst-case (*i.e.*, minimum) *angle* between the data points and the separating hyperplane.

This perspective applies to the general problem (7.68) with nonzero b (with careful notation and interpretation).

Figure 7.10 shows an example of the maximum weight error margin linear discrimination on a randomly generated dataset in $\mathbf{R}^2$. The separating hyperplane $\{z \in \mathbf{R}^n \mid a^{\star T} z = b^\star\}$ (shown solid) is defined by the optimal point $(a^\star, b^\star, t^\star)$ of the problem (7.68), where the optimal value $t^\star$ corresponds to the weight error margin of this separating hyperplane. The shaded region represents the set given by

$$\{z \in \mathbf{R}^n \mid -t^\star \|z\|_2 \leq a^{\star T} z - b^\star \leq t^\star \|z\|_2\}, \tag{7.70}$$

which corresponds to the set of all separating hyperplanes in the form (7.65) that can be obtained by perturbing the weight vector $a^\star$ with $w \in \mathbf{R}^n$ within the weight error margin, *i.e.*, $\|w\|_2 \leq t^\star$. Geometrically, the set given by (7.70) forms a banded region bounded by two quadratic surfaces.

7.4 Support vector classifiers

When the two groups of points $\{x_1, \ldots, x_M\}$ and $\{y_1, \ldots, y_N\}$ in $\mathbf{R}^n$ are not linearly separable, we might still want to find an affine function $f\colon \mathbf{R}^n \to \mathbf{R}$ of the form $f(x) = a^T x - b$ that can approximately separate the two groups of points.

We start from the linear discrimination feasibility problem (7.53), which is infeasible when the two groups of points are not linearly separable. In this case, we may introduce auxiliary variables $u \in \mathbf{R}^M$ and $v \in \mathbf{R}^N$ to relax the constraints in (7.53), so that we have the problem

$$\begin{array}{ll} \text{find} & (a, b, u, v) \\ \text{subject to} & a^T x_i - b \geq 1 - u_i, \quad i = 1, \ldots, M \\ & a^T y_i - b \leq -1 + v_i, \quad i = 1, \ldots, N \\ & u \succeq 0, \quad v \succeq 0, \end{array} \tag{7.71}$$

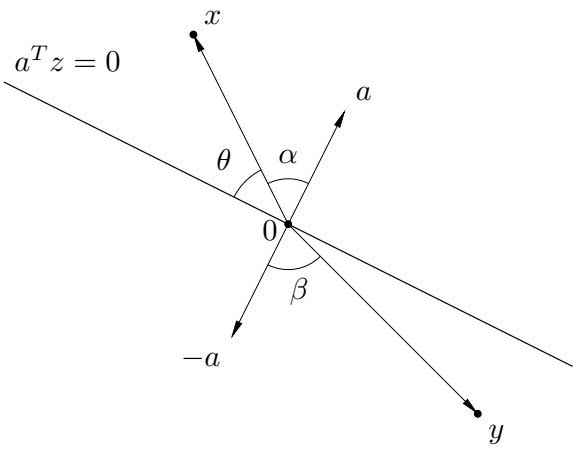

Figure 7.9 The points x and y are two data points from different groups in the problem (7.69), which are separated by the hyperplane $\{z \mid a^T z = 0\}$. The angle α is the angle between the point x and the normal vector a of the separating hyperplane, and the angle β is the angle between the point y and the normal vector $-a$ of the same separating hyperplane. The angle θ is the angle between the data point (either x or y, only shown for x) and the separating hyperplane itself, which is the complement of the angle α or β.

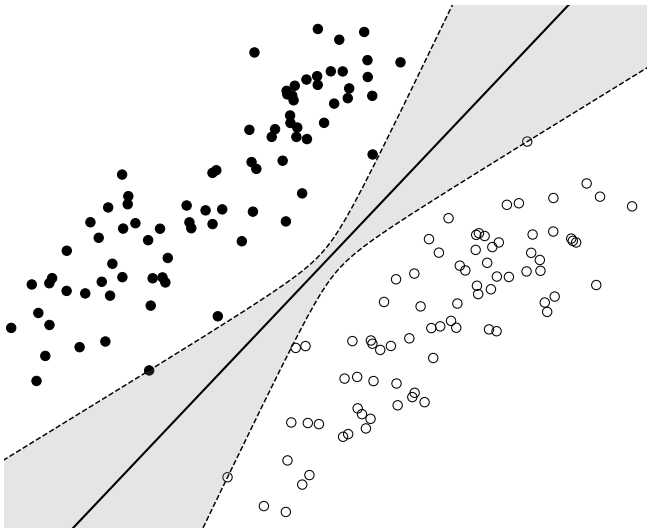

Figure 7.10 The dots and circles represent two groups of points in $\mathbf{R}^2$ that are linearly separable. The solid line represents the separating hyperplane that solves the problem (7.68). The shaded region shows the set (7.70), which corresponds to the set of all separating hyperplanes that can be obtained by perturbing the normal vector of the separating hyperplane within the weight error margin.

where $a \in \mathbf{R}^n$, $b \in \mathbf{R}$, $u \in \mathbf{R}^M$, and $v \in \mathbf{R}^N$ are the optimization variables.

The relaxed problem (7.71) is always feasible, since for any choice of a and b, we can always make the auxiliary variables u and v sufficiently large so that the constraints are satisfied. In this case, for fixed values of a and b, the smallest values of u and v that make the separation constraints feasible can be interpreted as the *margin violations* or *overlap* of the two groups of points in $\{x_1, \ldots, x_M\}$ and $\{y_1, \ldots, y_N\}$, respectively, with respect to the 'separating slab' defined by

$$\{z \in \mathbf{R}^n \mid -1 \leq a^T z - b \leq 1\}, \tag{7.72}$$

whose boundary consists of two parallel hyperplanes

$$\{z \in \mathbf{R}^n \mid a^T z = b - 1\} \quad \text{and} \quad \{z \in \mathbf{R}^n \mid a^T z = b + 1\}. \tag{7.73}$$

Note that if the smallest feasible u_i for some $x_i \in \{x_1, \ldots, x_M\}$ is nonzero but less than 1, then the point x_i stays within the slab, but is not misclassified, since we still have $a^T x_i - b \geq 1 - u_i > 0$. If $u_i \geq 1$, then the point x_i is misclassified, since in this case $a^T x_i - b < 0$. We have similar results for the variables v_i and the points y_i. If the two groups of points are linearly separable, then there is a separating hyperplane such that the margin violation of both groups is zero, *i.e.*, the problem (7.71) is feasible with $u = 0$ and $v = 0$.

The next question is how to choose a separating hyperplane based on the problem (7.71) so that the margin violation is in some sense small, which forms the class of *support vector classifiers* (or *support vector machine*).

7.4.1 Margin violation penalties

The basic idea of support vector classifiers is to choose the separating hyperplane by minimizing some penalty function of the margin violations u and v, based on the problem (7.71). We discuss several different choices of the penalty function in the following paragraphs.

Least squares penalty

One natural approach to refine the problem (7.71) is to minimize the sum of the squares of the margin violations, which leads to the following optimization problem:

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^M u_i^2 + \sum_{i=1}^N v_i^2 \\ \text{subject to} & a^T x_i - b \geq 1 - u_i, \quad i = 1, \ldots, M \\ & a^T y_i - b \leq -1 + v_i, \quad i = 1, \ldots, N \\ & u \succeq 0, \quad v \succeq 0 \end{array} \tag{7.74}$$

with variables $a \in \mathbf{R}^n$, $b \in \mathbf{R}$, $u \in \mathbf{R}^M$, and $v \in \mathbf{R}^N$. This problem is a convex optimization problem with convex quadratic objective and linear inequality constraints.

Since the objective of the problem (7.74) corresponds to adding a least squares (or quadratic) penalty on the margin violations, the problem (7.74) is sometimes called the *least squares support vector classifier*. As a result of the quadratic penalty

function, this type of classifier is expected to find a separating hyperplane that prevents large margin violations, but there could still be many points that are misclassified or within the separating slab.

Figure 7.11 illustrates the separating hyperplane obtained by solving the problem (7.74) on a randomly generated dataset in $\mathbf{R}^2$. The points in $\{x_1, \ldots, x_M\}$ are represented by circles, and the points in $\{y_1, \ldots, y_N\}$ are represented by dots. The solid line represents the separating hyperplane, while the separating slab defined by (7.72) is shown shaded. It is observed that there are 2 misclassified points in the group $\{x_1, \ldots, x_M\}$ (*i.e.*, the circles on the left-hand side of the separating hyperplane), and there are 3 misclassified points in the group $\{y_1, \ldots, y_N\}$ (*i.e.*, the dots on the right-hand side of the separating hyperplane). There are also several points that are within the separating slab but are not misclassified. The data points that are on outside the shaded region are all correctly classified. In particular, these points are strictly separated by the separating slab (and, of course, strictly separated by the separating hyperplane).

Sparse margin violations

Another type of support vector classifiers is to minimize the total number of points that violate the separation constraints in (7.71), which leads to the cardinality minimization problem

$$\begin{array}{ll} \text{minimize} & \mathbf{card}\, u + \mathbf{card}\, v \\ \text{subject to} & a^T x_i - b \geq 1 - u_i, \quad i = 1, \ldots, M \\ & a^T y_i - b \leq -1 + v_i, \quad i = 1, \ldots, N \\ & u \succeq 0, \quad v \succeq 0 \end{array} \tag{7.75}$$

with variables $a \in \mathbf{R}^n$, $b \in \mathbf{R}$, $u \in \mathbf{R}^M$, and $v \in \mathbf{R}^N$, where the cardinality functions $\mathbf{card}\, u$ and $\mathbf{card}\, v$ denote the number of nonzero entries in u and v, respectively. The problem (7.75) is a very hard combinatorial optimization problem, which can become computationally intractable when the total number of points $M + N$ is large.

In practice, the problem (7.75) is often approximately solved via the ℓ_1-norm heuristic, which reduces to solving the linear program

$$\begin{array}{ll} \text{minimize} & \mathbf{1}^T u + \mathbf{1}^T v \\ \text{subject to} & a^T x_i - b \geq 1 - u_i, \quad i = 1, \ldots, M \\ & a^T y_i - b \leq -1 + v_i, \quad i = 1, \ldots, N \\ & u \succeq 0, \quad v \succeq 0. \end{array} \tag{7.76}$$

Note that the variables u and v are nonnegative, so the ℓ_1-norm of these vectors is equivalent to the sum of their entries, as shown by the linear objective above.

Figure 7.12 illustrates the separating hyperplane obtained by solving the problem (7.76) on the same dataset shown in figure 7.11. Compared to the results from the least squares support vector classifier via the problem (7.74), the separating hyperplane obtained by solving the problem (7.76) results in the same misclassified

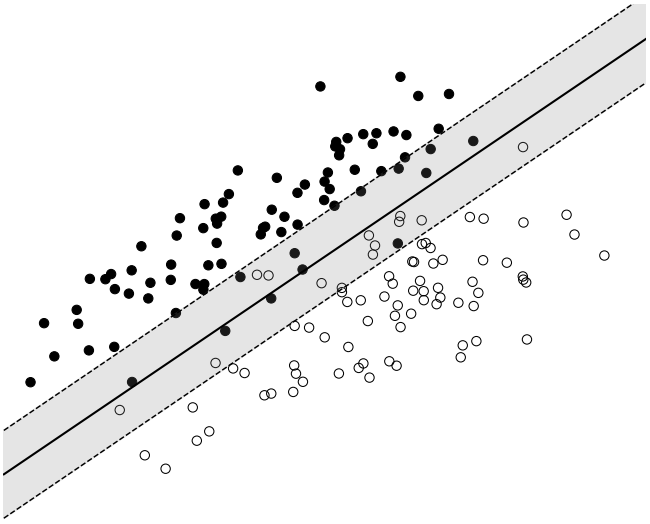

Figure 7.11 The circles and dots represent two groups of points $\{x_1, \ldots, x_M\}$ and $\{y_1, \ldots, y_N\}$ in $\mathbf{R}^2$, respectively, which are not linearly separable. The solid line represents the separating hyperplane obtained by solving the problem (7.74), and the shaded region represents the separating slab with respect to this separating hyperplane, where its boundaries (shown dashed) are defined by (7.73).

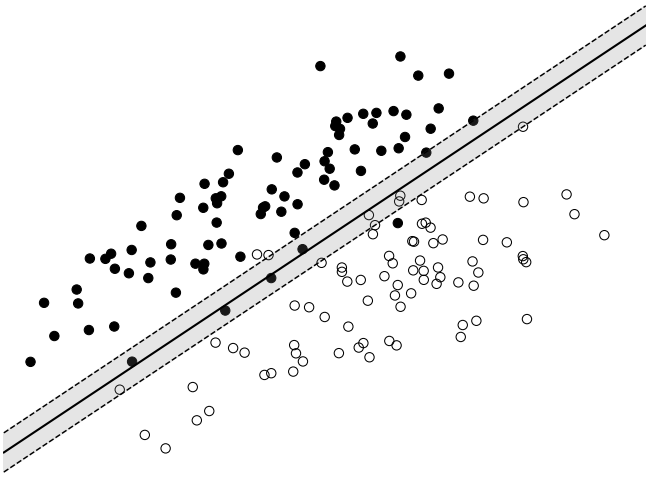

Figure 7.12 Separating hyperplane (shown solid) and the corresponding slab (shown shaded) obtained by solving the problem (7.76) on the same dataset shown in figure 7.11. The boundaries of the separating slab are defined by (7.73) and are shown dashed.

points, but the total number of points that are within the separating slab is significantly reduced. On the other hand, there are three misclassified points stay even outside the separating slab, which correspond to margin violations that are larger than 2. This is because minimizing the ℓ_1-norm of the vectors u and v encourages the margin violations to be sparse, so the separating hyperplane obtained by solving the problem (7.76) is expected to have only a few points that are misclassified or within the separating slab, where the compromise then is that the maximum value of the margin violations might be large.

7.4.2 Standard form support vector classifier

The standard form of support vector classifier takes the ℓ_1-norm as the penalty function and additionally considers the trade-off between the margin violation and the margin width (*i.e.*, the separating slab thickness). In particular, it solves the following optimization problem:

$$\begin{array}{ll} \text{minimize} & \|a\|_2 + \gamma(\mathbf{1}^T u + \mathbf{1}^T v) \\ \text{subject to} & a^T x_i - b \geq 1 - u_i, \quad i = 1, \ldots, M \\ & a^T y_i - b \leq -1 + v_i, \quad i = 1, \ldots, N \\ & u \succeq 0, \quad v \succeq 0, \end{array} \tag{7.77}$$

where $a \in \mathbf{R}^n$, $b \in \mathbf{R}$, $u \in \mathbf{R}^M$, and $v \in \mathbf{R}^N$ are the optimization variables, and $\gamma > 0$ is a penalty parameter that controls the trade-off between the margin violation and the margin width.

Notice that the thickness of the separating slab defined by (7.72) is equal to $2/\|a\|_2$ (see also remark 7.3), so minimizing the first term $\|a\|_2$ in the objective of (7.77) is equivalent to maximizing the margin width. The second term $\mathbf{1}^T u + \mathbf{1}^T v$ in the objective of (7.77) is equivalent to the objective of the problem (7.76), which encourages the margin violations to be sparse. Therefore, the coefficient $\gamma > 0$ can be interpreted as the relative weight of the total number of points that violate the separation conditions (which we want to minimize), compared to the thickness of the separating slab (which we want to maximize). In particular, when γ is very large, the problem (7.77) is expected to find a separating hyperplane that has very few margin violations, which might result in a small margin width; when γ is very small, the problem (7.77) is expected to find a separating hyperplane that has a large margin width, which might result in many margin violations. Under this interpretation, the problem (7.76) can be viewed as a special case of the problem (7.77) with $\gamma \to \infty$, where the margin width is not considered and the margin violation is the only concern.

Figure 7.13 and figure 7.14 illustrate the separating hyperplanes obtained by solving the problem (7.77) with $\gamma = 0.01$ and $\gamma = 1$, respectively, on a randomly generated dataset in $\mathbf{R}^2$. When $\gamma = 0.01$, the separating hyperplane leads to a large margin width, but there are many points that are misclassified or within the separating slab. When taking a larger $\gamma = 1$, the thickness of the separating slab decreases, with the advantage that there are only a few points that are misclassified or within the separating slab.

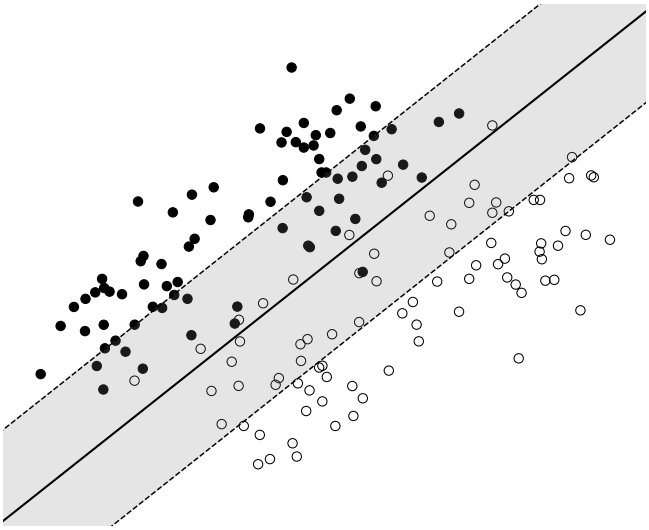

Figure 7.13 The circles and dots represent two groups of points that are not linearly separable. The solid line represents the separating hyperplane obtained by solving the problem (7.77) with $\gamma = 0.01$, and the shaded region represents the separating slab with respect to this separating hyperplane, where its boundaries (shown dashed) are defined by (7.73).

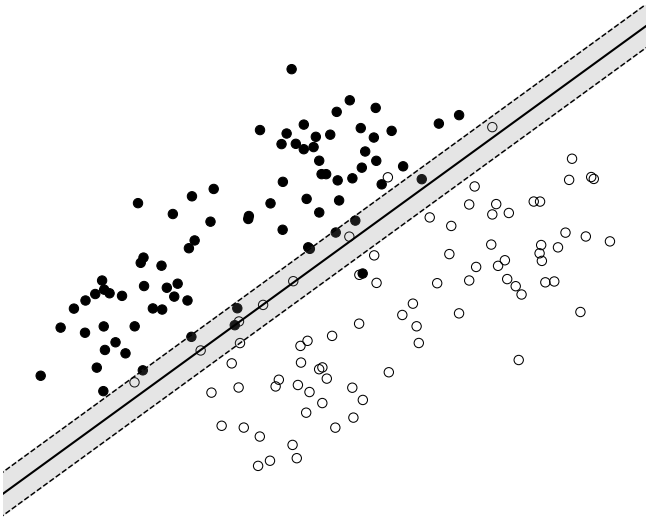

Figure 7.14 Separating hyperplane (shown solid) and the corresponding slab (shown shaded) obtained by solving the problem (7.77) with $\gamma = 1$ on the same dataset shown in figure 7.13.

Bibliographical notes

Stochastic programming and chance constrained problems are discussed in many textbooks, such as Prékopa [Pré95], Birge and Louveaux [BL11], Shapiro *et al.* [SDR21]. Algorithms for stochastic optimization and some related applications can be found in the books by Uryasev and Pardalos (editors) [UP01] and Wallace and Ziemba [WZ05].

For some theoretical results regarding the convergence of the sample average approximation method for solving stochastic programming problems, see [SDR21, chapter 5, theorem 5.3, proposition 5.6].

Chance constrained problems were first introduced by Charnes and Cooper in 1959 [CC59], and were later developed by Miller and Wagner in 1965 [MW65]. The convex conservative approximations of chance constraints discussed in §7.1.2, page 256, were introduced by Nemirovski and Shapiro [NS06]. They showed that the tightest possible choice of $\phi\colon \mathbf{R} \to \mathbf{R}$ in (7.21) corresponds to the Markov approximation discussed in example 7.7, which can also be interpreted in terms of the *conditional value-at-risk* of $f(x, \omega)$; see also Rockafellar and Uryasev [RU02].

Some early papers on worst-case robust convex optimization include El Ghaoui and Lebret [EL97], Ben-Tal and Nemirovski [BN98, BN99], El Ghaoui *et al.* [EOL98], and Goldfarb and Iyengar [GI03]. These ideas were later summarized in the book by Ben-Tal *et al.* [BEN09]. See also Boyd and Vandenberghe [BV04, §4.4.2] for some examples. Some recent reviews on robust optimization can be found in the papers by Ben-Tal and Nemirovski [BN02], Beyer and Sendhoff [BS07], Bertsimas *et al.* [BBC11], and Gabrel *et al.* [GMT14].

Applications of worst-case robust optimization in linear approximation problems originate from [EL97] and [CGGS98], and were later summarized in [BV04, §6.4]. The worst-case robust approximation problem (7.42) with uncertainty set (7.46) for $k > 1$ can be solved via semidefinite programming; see [EL97]. A variation of this problem with the ℓ_∞-norm in the approximation error and uncertainty set is discussed in [BV04, page 323].

Practical linear discrimination via convex optimization (and extensions to nonlinear discrimination) was introduced by Mangasarian [Man65] and Rosen [Ros65] in the 1960s. Linear discrimination robust to weight vector perturbations discussed in §7.3.3 is adapted from [BV04, exercise 8.24], which can be considered as a special case of the robust classification problem presented in Ben-Tal *et al.* [BEN09, §12.1.1, page 302]. See also the bibliographical notes of chapter 4 for more references on the theoretical foundations of discrimination feasibility.

The original idea of support vector classifiers for linear and nonlinear discrimination has been introduced by Vapnik and Chervonenkis [VC64] since the 1960s in the Soviet Union, and was later popularized by Boser *et al.* [BGV92] and Cortes and Vapnik [CV95] in the 1990s. For more details about the support vector classifiers, see the books by Vapnik [Vap98, Vap00] and Schölkopf and Smola [SS01, part II]. A brief history about the development of support vector classifiers in the early stages can be found in the afterword, chapter 2, of the book [Vap06]. These topics are also discussed in many textbooks on machine learning and pattern recognition, such as Duda *et al.* [DHS00, chapter 5], Hastie *et al.* [HTF09, §4.5 and chapter 12], Bishop [Bis06, chapter 7], Murphy [Mur12, §14.5], and Murphy [Mur22, §17.3 and §17.4].

The least squares support vector classifier discussed on page 283 was introduced by Suykens and Vandewalle [SV99]. The idea of chance constraints can also be used to formulate support vector classifiers; see Ben-Tal *et al.* [BBBS11].

Exercises

Stochastic optimization

7.1 *Chance constraint with log-concave distribution.* Let $f\colon \mathbf{R}^n \times \mathbf{R}^q \to \mathbf{R}$ be a function with variable $x \in \mathbf{R}^n$ and random parameter $\omega \in \mathbf{R}^q$. Assuming that

- the function f is jointly convex in x and ω, and
- the random parameter ω has a log-concave distribution $p(\omega)$,

we want to show that the chance constraint

$$\mathbf{prob}(f(x,\omega) \leq 0) \geq \eta$$

can be expressed as a convex constraint in x.

(a) We first define the function $g\colon \mathbf{R}^n \times \mathbf{R}^q$ as

$$g(x,\omega) = \begin{cases} 1, & f(x,\omega) \leq 0 \\ 0, & \text{otherwise.} \end{cases}$$

Show that the function g is log-concave in (x,ω).
Hint. Recall (from exercise 2.11) that a function $h\colon \mathbf{R}^m \to \mathbf{R}$ with convex domain and $h(x) > 0$ for all $x \in \mathbf{dom}\, f$ is log-concave if and only if

$$h(\theta x + (1-\theta)y) \geq h(x)^\theta h(y)^{1-\theta}$$

for all $x, y \in \mathbf{dom}\, h$ and $\theta \in [0,1]$.

(b) If a function $h\colon \mathbf{R}^m \times \mathbf{R}^k \to \mathbf{R}$ is log-concave (jointly in its two arguments), then the function $H\colon \mathbf{R}^m \to \mathbf{R}$ defined by the integration

$$H(x) = \int h(x,y)\, dy$$

is a log-concave function of x (where the integration is over $\mathbf{R}^k$) [Pré71, Pré73, Pré80]. Use this result to show that the probability $\mathbf{prob}(f(x,\omega) \leq 0)$ is a log-concave function of x.

(c) Formulate the chance constraint $\mathbf{prob}(f(x,\omega) \leq 0) \geq \eta$ as a convex constraint of x.

7.2 [Boy11, page 16] *Traditional Chebyshev chance constraint bound.* Consider the chance constraint given by $\mathbf{prob}(f(x,\omega) \leq 0) \geq \eta$ with $\eta \in (0,1)$, where $f\colon \mathbf{R}^n \times \mathbf{R}^q \to \mathbf{R}$ is a function with variable $x \in \mathbf{R}^n$ and random parameter $\omega \in \mathbf{R}^q$.

(a) Show that we have the following bound on the probability $\mathbf{prob}(f(x,\omega) > 0)$:

$$\mathbf{prob}(f(x,\omega) > 0) \leq \mathbf{E}\,(f(x,\omega)/\alpha + 1)^2$$

where $\alpha > 0$ is any positive scalar. This bound is sometimes called the *traditional* Chebyshev bound for chance constraints, to distinguish it from the Chebyshev bound in (7.28).

(b) The conservative approximation of the chance constraint $\mathbf{prob}(f(x,\omega) \leq 0) \geq \eta$ based on the traditional Chebyshev bound is given by

$$\mathbf{E}\,\alpha(f(x,\omega)/\alpha + 1)^2 \leq \alpha(1-\eta), \tag{7.78}$$

which can be expressed as

$$2\,\mathbf{E}\,f(x,\omega) + (1/\alpha)\,\mathbf{E}\,f(x,\omega)^2 + \alpha\eta \leq 0. \tag{7.79}$$

i. Is the inequality (7.79) a convex constraint jointly in x and α? If not, what kind of assumption on the function f would make it convex?

ii. How could choose α to make the approximation (7.79) as tight as possible? What is the tightest approximation we can get? What about its convexity properties? Is there any requirement on the distribution of $f(x,\omega)$ to use this tightest approximation?
Hint. One way to see the convexity of the tightest approximation is via partial minimization of a convex function; see exercise 2.12.

(c) The approximation (7.78) could also be expressed as

$$\eta\alpha^2 + 2\,\mathbf{E}\,f(x,\omega)\alpha + \mathbf{E}\,f(x,\omega)^2 \leq 0. \tag{7.80}$$

Answer the same questions i. and ii. from (b), for the approximation (7.80).

Worst-case optimization

7.3 *Robust approximation with convex hull of finite matrices uncertainty set.* Consider the worst-case robust approximation problem

$$\text{minimize} \quad \sup\{\|Ax-b\| \mid A \in \mathcal{A}\},$$

where $x \in \mathbf{R}^n$ is the optimization variable, $b \in \mathbf{R}^m$ is a given fixed vector, and the feature matrix $A \in \mathbf{R}^{m\times n}$ is uncertain and belongs to the uncertainty set $\mathcal{A} \subseteq \mathbf{R}^{m\times n}$. The norm $\|\cdot\|$ can be any norm on $\mathbf{R}^m$. Here we assume the uncertainty set $\mathcal{A}$ is given by

$$\mathcal{A} = \mathbf{conv}\{A_1,\ldots,A_k\},$$

where the matrices $A_1,\ldots,A_k \in \mathbf{R}^{m\times n}$ are given. Express this problem as a convex optimization problem. When does the problem becomes a linear program?

7.4 [EL97, CGGS98] *Robust approximation with norm ball uncertainty set.* Consider the worst-case robust approximation problem

$$\text{minimize} \quad \sup\{\|Ax-b\|_2 \mid A \in \mathcal{A}\}, \tag{7.81}$$

where $x \in \mathbf{R}^n$ is the optimization variable, $b \in \mathbf{R}^m$ is a given fixed vector, and the feature matrix $A \in \mathbf{R}^{m\times n}$ is uncertain and belongs to the uncertainty set

$$\mathcal{A} = \{\bar{A} + U \mid \|U\|_2 \leq \gamma\}, \tag{7.82}$$

where $\bar{A} \in \mathbf{R}^{m\times n}$ is a given nominal matrix and $\gamma > 0$ is a given norm bound on the uncertainty matrix $U \in \mathbf{R}^{m\times n}$. We should carefully parse the norm $\|\cdot\|_2$ in (7.81) and (7.82), where the first one is the Euclidean norm on $\mathbf{R}^m$ and the second one is the spectral norm on $\mathbf{R}^{m\times n}$. Express the problem (7.81) as a convex optimization problem. You need to evaluate the inner supremum in the problem (7.81) to get a closed-form expression of the objective function.

Robust discrimination and support vector classifiers

7.5 Consider the maximum margin linear discrimination problem (7.56). Show that at an optimal point $(\tilde{a}^\star, \tilde{b}^\star, t^\star)$ of the problem (7.56), there must be at least one index $i \in \{1,\ldots,M\}$ such that $\tilde{a}^{\star T}x_i - \tilde{b}^\star = t^\star$, and at least one index $i \in \{1,\ldots,N\}$ such that $\tilde{a}^{\star T}y_i - \tilde{b}^\star = -t^\star$. In other words, the optimal value $t^\star$ of the problem (7.56) is equal to the distance from the separating hyperplane $\{z \in \mathbf{R}^n \mid \tilde{a}^{\star T}z = \tilde{b}^\star\}$ to the closest point in both groups of points $\{x_1,\ldots,x_M\}$ and $\{y_1,\ldots,y_N\}$.

7.6 Consider the maximum weight error margin linear discrimination problem (7.68). Suppose the given two groups of points $\{x_1, \ldots, x_M\}$ and $\{y_1, \ldots, y_N\}$ are linearly separable. Show that the problem (7.68) is equivalent to the following problem:

$$\begin{array}{ll} \text{minimize} & \|\tilde{a}\|_2 \\ \text{subject to} & \tilde{a}^T x_i - \tilde{b} \geq \|x_i\|_2, \quad i = 1, \ldots, M \\ & \tilde{a}^T y_i - \tilde{b} \leq -\|y_i\|_2, \quad i = 1, \ldots, N \end{array}$$

with variables $\tilde{a} \in \mathbf{R}^n$ and $\tilde{b} \in \mathbf{R}$.
Hint. Using the change of variables $\tilde{a} = a/t$ and $\tilde{b} = b/t$ with $t > 0$.

7.7 *Worst-case margin violation.* Consider feasibility problem (7.71) corresponding to the support vector classifier. Suppose we want to find a separating hyperplane that minimizes the worst-case margin violation, which is given by

$$\max_{\substack{i=1,\ldots,M \\ j=1,\ldots,N}} \{u_i, v_j\} = \max\{\|u\|_\infty, \|v\|_\infty\}.$$

(a) Formulate this problem as an LP.

(b) What does a solution of this worst-case margin violation minimization problem look like? Consider the cases when the two groups of points $\{x_1, \ldots, x_M\}$ and $\{y_1, \ldots, y_N\}$ are linearly separable and when they are not linearly separable.

(c) Is this problem formulation useful in practice? If not, can you think of any modifications to make it more useful? You should not formulate the modified problem as a nonlinear program.

Chapter 8

Latent factor estimation

8.1 Mixture models

8.1.1 The inverse problem

Let (x_i, y_i), $i = 1, \ldots, m$, be a dataset of m samples, where $x_i \in \mathbf{R}^n$ are the feature vectors and $y_i \in \mathbf{R}$ are the corresponding responses. The following inverse problem fits a general machine learning model to this dataset:

$$\text{minimize} \quad \sum_{i=1}^m f_\theta(x_i, y_i) \tag{8.1}$$

where the variable $\theta \in \mathbf{R}^n$ represents the model parameters, and the function $f_\theta \colon \mathbf{R}^n \times \mathbf{R} \to \mathbf{R}$ is a cost function that measures the error of fitting each data point (x_i, y_i) with the model parameterized by θ.

A *mixture model*, or *hierarchical model*, extends the basic machine learning model with inverse problem (8.1) by assuming that there are multiple models that can explain the data, and that each data point is generated by one of these models. Specifically, let $\theta_1, \ldots, \theta_k \in \mathbf{R}^n$ be the parameters of k different models, and let $z_i \in \{e_1, \ldots, e_k\} \subseteq \mathbf{R}^k$ be the *one-hot* encoded label of the model that generates the data point (x_i, y_i), where e_j is the jth standard basis vector in $\mathbf{R}^k$. The inverse problem corresponding to this mixture model is then given by

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^m \sum_{j=1}^k z_{ij} f_{\theta_j}(x_i, y_i) \\ \text{subject to} & z_i \in \{e_1, \ldots, e_k\}, \quad i = 1, \ldots, m, \end{array} \tag{8.2}$$

where $z_i \in \mathbf{R}^k$ for $i = 1, \ldots, m$ (with z_{ij} being its jth component) and $\theta_1, \ldots, \theta_k \in \mathbf{R}^n$ are the variables of the problem. The one-hot labels $z_1, \ldots, z_m$ are sometimes called the (discrete) *hidden* or *latent factors* of this mixture model, since they are not observed in the dataset and need to be inferred from the data.

Note that for simplicity of presentation, in the following discussion we assume that the cost functions f_{θ_i} for $i = 1, \ldots, k$ in (8.2) share the same functional form, but have different parameters $\theta_i \in \mathbf{R}^n$. These results are readily extended to the general case where the cost functions might have different functional forms.

Interpretation

We can rewrite the problem (8.2) to better understand its structure:

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^m z_i^T r_i \\ \text{subject to} & r_{ij} = f_{\theta_j}(x_i, y_i), \quad i = 1, \ldots, m, \quad j = 1, \ldots, k \\ & z_i \in \{0,1\}^k, \quad \mathbf{card}\, z_i = 1, \quad i = 1, \ldots, m, \end{array} \tag{8.3}$$

where $r_i \in \mathbf{R}^k$ for $i = 1, \ldots, m$ are auxiliary variables. The variables r_i can be interpreted as the vector of costs, or errors, of fitting the data point (x_i, y_i) with the k different models. The constraints on z_i for $i = 1, \ldots, m$ enforce z_i are standard basis vectors, so each data point is assigned to exactly one model. Therefore, for the ith data point, the value $z_i^T r_i$ is essentially the cost of fitting that data point with the model assigned to it by z_i, and our goal is to find the model parameters $\theta_1, \ldots, \theta_k$ and the assignment variables $z_1, \ldots, z_m$ that minimize the total cost of fitting all data points.

From a hierarchical modeling perspective, we can interpret the problem (8.3) as follows. By solving the problem (8.3), we first separate the m data points into k different groups, according to the assignment given by the latent factors $z_1, \ldots, z_m$, and then fit a model with parameter θ_i to the data points in the ith group, individually for $i = 1, \ldots, k$, so that the model fitting error of all groups is minimized.

8.1.2 Relaxation and biconvex formulation

It is easily seen that the problem (8.3) is not a convex optimization problem. In fact, it is NP-hard in general, even when the cost functions f_{θ_i} for $i = 1, \ldots, k$ are convex in θ_i.

Remark 8.1 *Proof of NP-hardness.* We reduce the well-known NP-hard problem of *clustering* to the problem (8.3). Consider the problem of clustering m data points $x_1, \ldots, x_m \in \mathbf{R}^n$ into k groups, where the goal is to find a partition of the data points into k disjoint subsets such that the total within-cluster variance (measured by the squared Euclidean distances to the cluster centers) is minimized. This problem can be formulated as follows:

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^m z_i^T r_i \\ \text{subject to} & r_{ij} = \|\theta_j - x_i\|_2^2, \quad i = 1, \ldots, m, \quad j = 1, \ldots, k \\ & z_i \in \{0,1\}^k, \quad \mathbf{card}\, z_i = 1, \quad i = 1, \ldots, m, \end{array} \tag{8.4}$$

where the variables $\theta_1, \ldots, \theta_k \in \mathbf{R}^n$ are the cluster centers, r_{ij} is the squared distance from the data point x_i to the cluster center θ_j, and $z_i \in \mathbf{R}^k$ is the one-hot assignment of the data point x_i to one of the k clusters for $i = 1, \ldots, m$. The problem (8.4) is a special case of the problem (8.3) with the cost functions $f_{\theta_i} : \mathbf{R}^n \times \mathbf{R} \to \mathbf{R}$ for $i = 1, \ldots, k$ defined as

$$f_{\theta_i}(x, y) = \|\theta_i - x - y\|_2^2,$$

and the dataset given by $(x_i, y_i = 0)$, $i = 1, \ldots, m$. This shows that the problem (8.3) is at least as hard as the clustering problem (8.4), and is therefore NP-hard.

We will discuss the clustering problem later in more detail in §8.2.

Suppose the cost functions f_{θ_i} for $i = 1, \ldots, k$ in the problem (8.3) are convex in θ_i, then the problem (8.3) is an integer constrained biconvex optimization problem, since the objective is convex in $\theta_1, \ldots, \theta_k$ when $z_1, \ldots, z_m$ are fixed, and is linear (and therefore convex) in $z_1, \ldots, z_m$ when $\theta_1, \ldots, \theta_k$ are fixed. Hence, when the dataset in (8.3) is not too large, *i.e.*, when m is small, this problem can be handled by enumerating all possible assignments of the data points $(x_1, y_1), \ldots, (x_m, y_m)$ to the k models. In particular, we can solve the convex optimization problem

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^m \tilde{z}_i^T r_i \\ \text{subject to} & r_{ij} = f_{\theta_j}(x_i, y_i), \quad i = 1, \ldots, m, \quad j = 1, \ldots, k, \end{array}$$

with variable $\theta_1, \ldots, \theta_k \in \mathbf{R}^n$, for the k^m possible choices of the latent factors $\tilde{z}_1, \ldots, \tilde{z}_m \in \mathbf{R}^k$ satisfying the constraints in (8.3), and then picking the one that gives the smallest objective value.

Heuristic via biconvex relaxation

In the most general case, when m is large, the problem (8.3) has to be solved approximately via some heuristics. To do this, we could first relax the integer constraints

$$z_i \in \{0, 1\}^k, \quad \mathbf{card}\, z_i = 1, \quad i = 1, \ldots, m,$$

to the convex constraints

$$0 \preceq z_i \preceq \mathbf{1}, \quad \mathbf{1}^T z_i = 1, \quad i = 1, \ldots, m,$$

so the problem (8.3) becomes

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^m z_i^T r_i \\ \text{subject to} & r_{ij} = f_{\theta_j}(x_i, y_i), \quad i = 1, \ldots, m, \quad j = 1, \ldots, k \\ & 0 \preceq z_i \preceq \mathbf{1}, \quad \mathbf{1}^T z_i = 1, \quad i = 1, \ldots, m, \end{array} \tag{8.5}$$

which is a biconvex optimization problem with variables $\theta_1, \ldots, \theta_k \in \mathbf{R}^n$ and $z_1, \ldots, z_m \in \mathbf{R}^k$ (assuming the functions f_{θ_i} are convex in θ_i for $i = 1, \ldots, k$).

Now to (approximately) solve (8.5), we can use the alternating minimization heuristic discussed in §3.1.3, which iterates between solving the convex subproblems

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^m z_i^T \tilde{r}_i \\ \text{subject to} & 0 \preceq z_i \preceq \mathbf{1}, \quad \mathbf{1}^T z_i = 1, \quad i = 1, \ldots, m, \end{array} \tag{8.6}$$

where $\tilde{r}_i = (f_{\tilde{\theta}_1}(x_i, y_i), \ldots, f_{\tilde{\theta}_k}(x_i, y_i))$ for $i = 1, \ldots, m$ are fixed problem data, and

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^m \tilde{z}_i^T r_i \\ \text{subject to} & r_{ij} = f_{\theta_j}(x_i, y_i), \quad i = 1, \ldots, m, \quad j = 1, \ldots, k, \end{array} \tag{8.7}$$

until the objective values of the two subproblems converge. Note that in the problem (8.6), the variables are $z_1, \ldots, z_m \in \mathbf{R}^k$, and the model parameters $\tilde{\theta}_1, \ldots, \tilde{\theta}_k \in \mathbf{R}^n$ are fixed to the values obtained from the previous solution of the problem (8.7), which, on the other hand, has variables $\theta_1, \ldots, \theta_k \in \mathbf{R}^n$, and the problem data $\tilde{z}_1, \ldots, \tilde{z}_m \in \mathbf{R}^k$ are fixed to the values obtained from the previous solution of the problem (8.6).

Interpretations

We can give the relaxation (8.5) the following interpretation. The original problem (8.3) constraints the latent factors $z_1, \ldots, z_m \in \mathbf{R}^k$ to be one of the standard basis vectors $e_1, \ldots, e_k \in \mathbf{R}^k$. Therefore, solving this problem consists in finding the optimal 'hard' assignment of the data points to the k models, where each data point is assigned to exactly one model. In this case, the cost of fitting each data point is only determined by the model assigned to it. The relaxed problem (8.5), however, allows the latent factors $z_1, \ldots, z_m$ to be any vectors in the convex hull of $\{e_1, \ldots, e_k\}$, *i.e.*, to stay within the probability simplex

$$\{z \in \mathbf{R}^k \mid 0 \preceq z \preceq \mathbf{1}, \quad \mathbf{1}^T z = 1\}.$$

Solving this problem therefore tries to find a 'soft' assignment of the data points to the k models, where the jth component of $z_i \in \mathbf{R}^k$ for $i = 1, \ldots, m$ can be interpreted as the probability of assigning the data point (x_i, y_i) to the jth model with parameter θ_j for $j = 1, \ldots, k$. Now the total cost of fitting the ith data point is the weighted average, or *expectation*, of the costs of fitting that data point with the k different models, where the expectation is taken with respect to the distribution given by the latent factor z_i.

We could also give the two subproblems (8.6) and (8.7) in the alternating minimization heuristic some interpretations. In the problem (8.6), the model parameters are fixed to $\tilde{\theta}_1, \ldots, \tilde{\theta}_k$, so the model fitting costs evaluated for each data point under all k models are fixed. Hence, the goal of this problem is to find the optimal assignment of the data points to the k models, so that each data point has larger assignment probability to the model that fits it better, and smaller assignment probability to the model that fits it worse, in order to minimize the total expected cost of fitting all data points. On the other hand, in the problem (8.7), the latent factors $\tilde{z}_1, \ldots, \tilde{z}_m$ are fixed, so each data point is assigned to the k models with fixed probabilities. In this case, solving the problem finds the optimal model parameters $\theta_1, \ldots, \theta_k$, where the parameter θ_i for the ith model tends to fit the data points that are assigned to it with larger probabilities better.

Remark 8.2 *Tightness of the probability relaxation.* In fact, solving the subproblem (8.6) under the probability simplex constraints on $z_1, \ldots, z_m \in \mathbf{R}^k$ for $i = 1, \ldots, m$ always gives a solution where each $z_i \in \{e_1, \ldots, e_k\}$ is a standard basis vector, given that there are no ties in the costs of fitting each data point with the k different models, *i.e.*, the vectors $\tilde{r}_1, \ldots, \tilde{r}_m \in \mathbf{R}^k$ for $i = 1, \ldots, m$ have unique minimum components. Intuitively, this is because if the vector $\tilde{r}_i$ has a unique minimum component, say the jth component, then an optimal point of the problem (8.6) must assign all the probability mass of z_i to the jth component, so that $z_i = e_j$ reduces to a standard basis vector in $\mathbf{R}^k$. (See also exercise 8.1 for a detailed proof of this result.)

Applying this observation to the problem (8.3) implies that its relaxation (8.5) is actually tight, *i.e.*, it always gives a solution where the latent factors $z_1, \ldots, z_m$ are one-hot encoded vectors, as long as there are no ties in the model fitting costs for each data point under the k different models at the optimal point. This does not always happen, though, and in these cases, the relaxation (8.5) can be useful in the sense that it is more convenient to incorporate regularization and constraints on the probability

distributions $z_1, \ldots, z_m \in \mathbf{R}^k$ (*e.g.*, those disciplined modules discussed in §4.3 and §6.3), so that we can break the ties with some prior knowledge; see also §8.1.4.

8.1.3 Cost functions

In this section, we introduce some example choices of the cost functions f_{θ_i} for $i = 1, \ldots, k$ in the problem (8.3) that lead to different types of mixture models.

Regression models

We could take the functions $f_{\theta_1}, \ldots, f_{\theta_k} : \mathbf{R}^n \times \mathbf{R} \to \mathbf{R}$ in the problem (8.3) to be of the form

$$f_{\theta_i}(x, y) = \phi(x^T\theta_i - y), \quad i = 1, \ldots, k, \tag{8.8}$$

where $\phi : \mathbf{R} \to \mathbf{R}$ is a penalty function. Examples of ϕ include, *e.g.*,

- the *quadratic penalty function* $\phi(u) = u^2$;
- the *ℓ_p-norm penalty function* $\phi(u) = |u|^p$ for $p \in [1, \infty)$ (see remark 4.2);
- the *Huber penalty function* $\phi(u) = u^2$ for $|u| \leq \delta$ and $\phi(u) = 2\delta|u| - \delta^2$ for $|u| > \delta$, where $\delta > 0$ is a parameter.

The basic problem (8.1) with cost function of the form (8.8) and quadratic penalty $\phi(u) = u^2$ consists in fitting a linear approximation or regression model to the dataset (x_i, y_i), $i = 1, \ldots, k$. Hence, the corresponding mixture model with inverse problem (8.3) is sometimes called a *mixture of linear regressions*. When the penalty function ϕ is chosen as the Huber penalty, the corresponding mixture model is sometimes called a *mixture of robust linear regressions*.

Example 8.1 *Mixture of linear regressions.* We consider a basic example of a mixture of linear regressions, corresponding to the inverse problem

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^m z_i^T r_i \\ \text{subject to} & r_{ij} = (x_i^T\theta_j - y_i)^2, \quad i = 1, \ldots, m, \quad j = 1, \ldots, k \\ & z_i \in \{0, 1\}^k, \quad \mathbf{card}\, z_i = 1, \quad i = 1, \ldots, m, \end{array} \tag{8.9}$$

where $\theta_1, \ldots, \theta_k \in \mathbf{R}^n$ and $z_1, \ldots, z_m \in \mathbf{R}^k$ are the optimization variables. We approximately solve this problem by the probability simplex relaxation and alternating minimization heuristic, where we iterate between solving the following two convex subproblems until convergence. The first subproblem is given by

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^m z_i^T \tilde{r}_i \\ \text{subject to} & 0 \preceq z_i \preceq \mathbf{1}, \quad \mathbf{1}^T z_i = 1, \quad i = 1, \ldots, m, \end{array}$$

where $z_1, \ldots, z_m \in \mathbf{R}^k$ are the variables and

$$\tilde{r}_i = \begin{bmatrix} (x_i^T\tilde{\theta}_1 - y_i)^2 \\ \vdots \\ (x_i^T\tilde{\theta}_k - y_i)^2 \end{bmatrix} \in \mathbf{R}^k, \quad i = 1, \ldots, m,$$

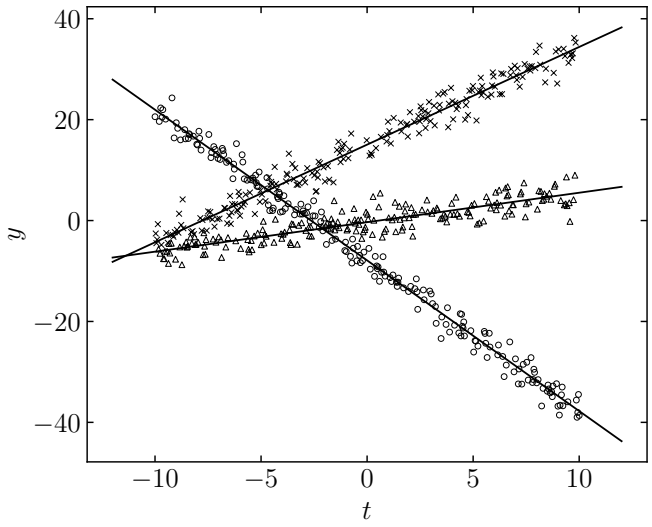

Figure 8.1 *Mixture of linear regressions.* Plot of the dataset used in example 8.1 and the fitted linear models (shown as lines). Each data point is shown with a marker corresponding to the group it is assigned to by the estimated latent factors $z_1, \ldots, z_m$ from solving (8.9).

are fixed problem data obtained from the previous solution of the second subproblem

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^m \tilde{z}_i^T r_i \\ \text{subject to} & r_{ij} = (x_i^T \theta_j - y_i)^2, \quad i = 1, \ldots, m, \quad j = 1, \ldots, k, \end{array}$$

where $\theta_1, \ldots, \theta_k \in \mathbf{R}^n$ are the variables and $\tilde{z}_1, \ldots, \tilde{z}_m \in \mathbf{R}^k$ are fixed problem data obtained from the previous solution of the first subproblem.

Figure 8.1 shows an example of fitting a mixture of linear regressions to a randomly generated dataset by approximately solving the problem (8.9) with the alternating minimization heuristic. The dataset (x_i, y_i), $i = 1, \ldots, m$, consists of $m = 600$ data points, where the feature vectors $x_i = (t_i, 1) \in \mathbf{R}^2$ for $i = 1, \ldots, m$, and the corresponding responses $y_i \in \mathbf{R}$ are generated by one of $k = 3$ different linear models, with additive noise. After solving the problem (8.9), the data points are partitioned into three groups according to the estimated latent factors $z_1, \ldots, z_m \in \mathbf{R}^k$, by assigning the ith data point to the jth group corresponding to the largest component of z_i for $i = 1, \ldots, m$, and are plotted in different markers. The fitted linear models are plotted as lines.

Classification models

We have seen in remark 8.1 that the clustering problem (8.4) corresponds to a special case of mixture models that can be used for classification. As another example, suppose we are given a dataset (x_i, y_i), $i = 1, \ldots, m$, where $x_i \in \mathbf{R}^n$ are the feature vectors and $y_i \in \{-1, 1\}$ are the corresponding binary class labels. We

could consider cost functions in the problem (8.3) given by

$$f_{\theta_i}(x, y) = \log(1 + \exp(-yx^T\theta_i)), \quad i = 1, \ldots, k, \tag{8.10}$$

which is essentially the negative log-likelihood of a *logistic model* (see §4.2.2). This cost function is sometimes called the *logistic loss function*, and the corresponding mixture model with inverse problem (8.3) is sometimes called a *hierarchical logistic regression*. Besides, we could consider the cost functions given by

$$f_{\theta_i}(x, y) = \max\{0,\ 1 - yx^T\theta_i\}, \quad i = 1, \ldots, k, \tag{8.11}$$

which is sometimes called the *hinge loss function*. Optimizing with respect to such cost function in (8.3) consists in fitting multiple *support vector classifiers* to the dataset (see §7.4.1, page 284). Furthermore, we may also consider the *exponential loss function* given by

$$f_{\theta_i}(x, y) = \exp(-yx^T\theta_i), \quad i = 1, \ldots, k, \tag{8.12}$$

which assigns exponentially larger cost to the data points that are misclassified by the model with parameter θ_i than to those that are correctly classified by it. In these cost functions, the term $yx^T\theta_i$ is called the *margin* of the data point (x, y) with respect to the model with parameter θ_i. All these costs are nonincreasing functions of the margin, so a larger margin of some data point corresponds to a smaller cost and is hence a better classification of the data by the model. The graphs of logistic, hinge, and exponential loss functions are shown in figure 8.2.

As a more complex example, we could generalize the logistic loss function to the *multiclass* setting. Suppose we are given a dataset $(X^{(i)}, y^{(i)})$, $i = 1, \ldots, m$, where

$$X^{(i)} = \begin{bmatrix} x_1^{(i)} & \cdots & x_q^{(i)} \end{bmatrix}^T \in \mathbf{R}^{q\times n}, \quad y^{(i)} \in \{e_1, \ldots, e_q\} \subseteq \mathbf{R}^q,$$

are the feature matrices and the corresponding one-hot class labels for the ith data point. In this context, the vector $x_j^{(i)} \in \mathbf{R}^n$ (whose transpose forms the jth row of the feature matrix $X^{(i)}$) is the feature vector contributing to the jth class, for $j = 1, \ldots, q$. For classification problems on this dataset, we could consider cost functions in the problem (8.3) that are of the form

$$f_{\theta_i}(X, y) = -\log(y^T u/\mathbf{1}^T u), \qquad u = \begin{bmatrix} \exp(x_1^T\theta_i) \\ \vdots \\ \exp(x_q^T\theta_i) \end{bmatrix}, \tag{8.13}$$

for $i = 1, \ldots, k$, where $x_1^T, \ldots, x_q^T$ are the rows of the matrix $X \in \mathbf{R}^{q\times n}$. The cost function given by (8.13) is essentially the negative log-likelihood of a *multiclass logistic model*, and is a convex function of θ_i (see exercise 4.4, page 141). The mixture model corresponding to the inverse problem (8.3) with cost function (8.13) is sometimes called a *hierarchical multiclass logistic regression*.

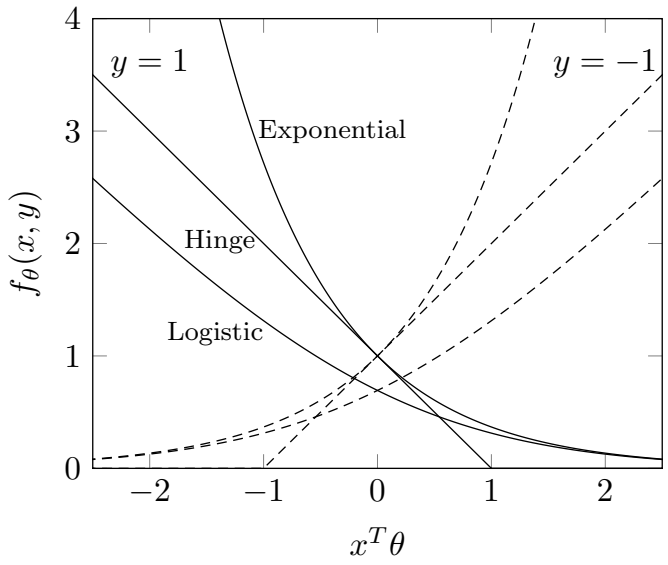

Figure 8.2 Graphs of the logistic, hinge, and exponential loss functions, given by (8.10), (8.11), and (8.12), respectively. The solid curves correspond to the cost function for the positive class ($y = 1$), and the dashed curves correspond to the cost function for the negative class ($y = -1$). The exponential loss assigns the largest cost to the misclassified data points.

8.1.4 Regularization and constraints

Sometimes we might want to add some regularization terms or constraints to the mixture model inverse problem (8.3) (or the relaxed problem (8.5)) to incorporate some prior knowledge about the latent factors $z_1, \ldots, z_m \in \mathbf{R}^k$ or the model parameters $\theta_1, \ldots, \theta_k \in \mathbf{R}^n$. Many disciplined modules that we have discussed in part II are readily adapted. For example, we could add the entropy regularization term

$$\gamma \sum_{i=1}^{m} \sum_{j=1}^{k} z_{ij} \log z_{ij},$$

where $\gamma > 0$ is a regularization parameter (see §4.3, page 134), to the problem (8.5) to encourage the distributions $z_1, \ldots, z_m \in \mathbf{R}^k$ over the k models to be as *stochastic* (*i.e.*, uniform) as possible, so that the data points are assigned to multiple models with more balanced probabilities. We give more ideas that are useful to the mixture model inverse problems in the next several paragraphs.

Parameter alignment

Note that the problem (8.3) is invariant to any permutation of the model parameters $\theta_1, \ldots, \theta_k$ and the corresponding latent factors $z_1, \ldots, z_m$. This is because any permutation of the model parameters and the corresponding latent factors does

not change the cost of fitting the data points, and hence does not change the objective value of the problem (8.3). Therefore, it is sometimes useful to add some constraints to the problem (8.3) to break this symmetry and align the model parameters $\theta_1, \ldots, \theta_k$ in some way (usually based on the prior knowledge), so that the solution of the problem is more interpretable.

Another potential application scenario where we might want to explicitly align the model parameters is when we have multiple datasets that are generated by the same set of models, and we want to fit a mixture model to each dataset. If the model parameters $\theta_1, \ldots, \theta_k$ are not aligned across different datasets, then the fitted models for different datasets might not be comparable, since the ith model for one dataset might correspond to the jth model for another dataset for some $i \neq j$.

Parameter alignment in fitting mixture models could be achieved by adding regularization terms and constraints that might be helpful in identifying the model parameters to the problem (8.3), based on the prior information about the models that generate the data. For example, if we know that the ith model with parameter θ_i is nonnegative, we could add the nonnegativity constraint $\theta_i \succeq 0$ to the problem (8.3). When fitting the mixture model multiple times or to different datasets, this restricts the nonnegative model parameters to be assigned to the same model across different fits. Similar techniques can be applied to incorporate prior information about, *e.g.*, monotonicity, sparsity, or smoothness properties in the model parameters. As long as the regularization terms and constraints added to the problem (8.3) are convex, the biconvex relaxation and alternating minimization heuristic on page 295 can still be applied to approximately solve the resulting inverse problem.

Time series

When the given dataset $(x(t), y(t))$, $t = 1, \ldots, m$, for the problem (8.3) is a time series, the latent factors $z(1), \ldots, z(m) \in \mathbf{R}^k$ to be estimated are often expected to be *piecewise constant*, since the data points in a time series are often generated by the same model for a contiguous period of time.

To achieve this required property, we could add the following regularization term to the problem (8.3):

$$\gamma \sum_{t=1}^{m-1} \|z(t+1) - z(t)\|_1, \tag{8.14}$$

where $\gamma > 0$ is a regularization parameter. This is essentially the *total variation smoothing* regularization (for time series of *vectors*; see also §5.3.3) applied to the latent factors $z(1), \ldots, z(m)$, which encourages the latent factors to be piecewise constant over time.

Example 8.2 *Hierarchical logistic regression for time series.* We consider a simple numerical example to illustrate the ideas discussed in the last several paragraphs. Suppose we are given a dataset $(x(t), y(t))$, $t = 1, \ldots, m$, where $x(t) \in \mathbf{R}^n$ are the feature vectors and $y(t) \in \{-1, 1\}$ are the corresponding responses at time t. We want to fit a mixture of $k = 2$ logistic regression models to this dataset, which corresponds to the inverse problem (8.3) with cost function given by (8.10). We want the latent factors $z(1), \ldots, z(m) \in \mathbf{R}^2$ to be piecewise constant over time, so we add the total

variation regularization term given by (8.14) to the problem (8.3). Additionally, we have the prior information that the model parameters $\theta_1 \in \mathbf{R}^n$ is nonnegative and monotone nonincreasing, *i.e.*,

$$\theta_1 \succeq 0, \qquad \theta_{1,1} \geq \cdots \geq \theta_{1,n},$$

and $\theta_2 \in \mathbf{R}^n$ is nonpositive and monotone nondecreasing in the same order, *i.e.*,

$$\theta_2 \preceq 0, \qquad \theta_{2,1} \leq \cdots \leq \theta_{2,n}.$$

Put together, the resulting (biconvex relaxed) inverse problem for this hierarchical logistic regression model is given by

$$\begin{array}{ll} \text{minimize} & \sum_{t=1}^{m} z(t)^T r(t) + \gamma \sum_{t=1}^{m-1} \|z(t+1) - z(t)\|_1 \\ \text{subject to} & r_i(t) = \log(1 + \exp(-y(t)x(t)^T\theta_i)), \quad t = 1, \ldots, m, \quad i = 1, 2 \\ & 0 \preceq z(t) \preceq \mathbf{1}, \quad \mathbf{1}^T z(t) = 1, \quad t = 1, \ldots, m \\ & \theta_1 \succeq 0, \quad \theta_{1,1} \geq \cdots \geq \theta_{1,n} \\ & \theta_2 \preceq 0, \quad \theta_{2,1} \leq \cdots \leq \theta_{2,n}, \end{array} \tag{8.15}$$

where $\theta_1, \theta_2 \in \mathbf{R}^n$, $z(1), \ldots, z(m) \in \mathbf{R}^2$ are the optimization variables, and $\gamma > 0$ is a regularization parameter.

Figure 8.3 shows an example of fitting this hierarchical logistic regression model to a randomly generated dataset with $n = 5$ and $m = 200$ by solving the biconvex problem (8.15) via alternate convex search, for different values of γ. The estimation from (8.15) and the ground truth are shown solid and dashed, respectively. The cross marks in the first row of the figure show the estimated group assignments $\hat{z}(1), \ldots, \hat{z}(m) \in \{1, 2\}$, *i.e.*, the largest component index of the estimated latent factors $z(t)$ for $t = 1, \ldots, m$, which is given by

$$\hat{z}(t) = \operatorname*{argmax}_{i \in \{1,2\}} z_i(t), \quad t = 1, \ldots, m. \tag{8.16}$$

When γ is small, *i.e.*, the total variation regularization is weak, the estimated group assignments are very noisy, and the data points are frequently assigned to different groups over time. On the other hand, when γ is large, the estimated group assignments shows a clear piecewise constant pattern, where the data points are assigned to the same group for a contiguous period of time. As a result, the estimated model parameters $\theta_1, \theta_2 \in \mathbf{R}^n$ are more close to the true model parameters (shown dashed) when γ is large than when γ is small, as shown in the bottom two rows of figure 8.3.

8.2 Clustering

8.2.1 The clustering problem

Suppose we have m data points $x_1, \ldots, x_m \in \mathbf{R}^n$, and we want to partition these data points into k groups or *clusters*, where k is a given positive integer. Usually, we have $k \ll m$ and we want the data points in the same cluster to be similar to each other, and the data points in different clusters to be different from each other as much as possible. Figure 8.4 shows a simple example in $\mathbf{R}^2$ with $m = 300$ data points and $k = 3$ clusters.

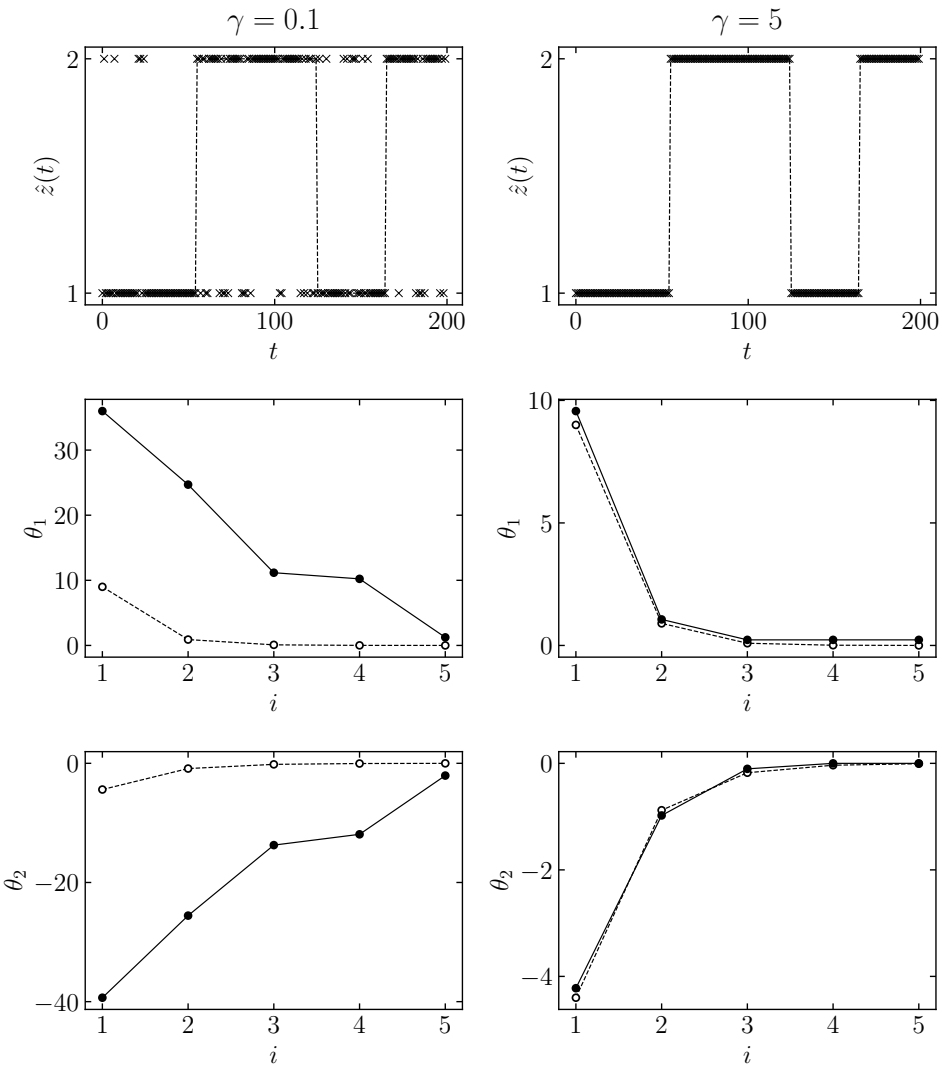

Figure 8.3 *Hierarchical logistic regression for time series.* Plot of the estimated group assignments $\hat{z}(1), \ldots, \hat{z}(m) \in \{1, 2\}$ given by (8.16) (shown as cross marks; first row) and the estimated model parameters $\theta_1, \theta_2 \in \mathbf{R}^n$ (shown solid; second and third rows) from solving the problem (8.15) with different values of the regularization parameter γ. The corresponding ground truth group assignments and model parameters are shown in dashed curves for comparison.

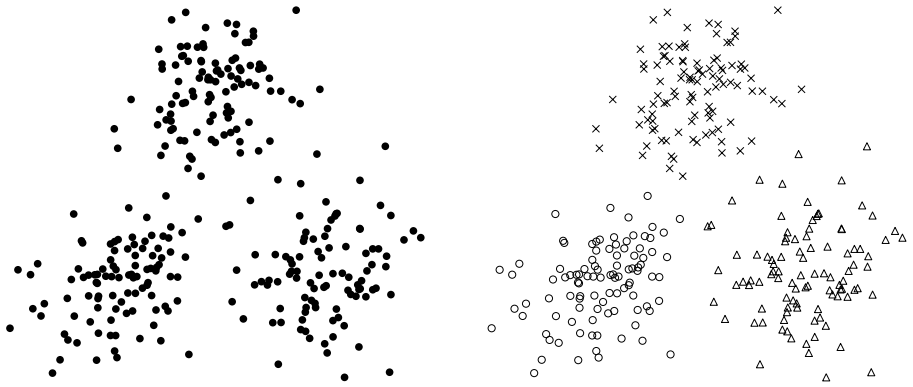

Figure 8.4 *Clustering.* Plot of 300 points in $\mathbf{R}^2$ (*left*, shown as dots) which can be clustered into 3 groups (*right*, shown in different markers).

A clustering objective

Let $\mu_1, \ldots, \mu_k \in \mathbf{R}^n$ be the *cluster representatives* for the k groups, then *a* formulation of the *clustering problem* is given by

$$\text{minimize} \quad \sum_{i=1}^m \min \left\{ \|x_i - \mu_j\|_2^2 \;\middle|\; j = 1, \ldots, k \right\} \tag{8.17}$$

where the optimization variables are $\mu_1, \ldots, \mu_k \in \mathbf{R}^n$. Geometrically, this clustering objective consists in finding k cluster representatives $\mu_1, \ldots, \mu_k \in \mathbf{R}^n$ so that the total (squared) Euclidean distance between the data points and their closest cluster representatives is as small as possible. In other words, the problem (8.17) consists in partitioning the data points $x_1, \ldots, x_m$ into k groups so that the data points in the same group are similar to each other as much as possible, in the sense that they are close to the same cluster representative.

We can rewrite the problem (8.17) more explicitly by introducing the cluster assignment variables $c \in \mathbf{R}^m$, where $c_i \in \{1, \ldots, k\}$ is the cluster index of the ith data point for $i = 1, \ldots, m$. Then the problem (8.17) is equivalent to

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^m \|x_i - \mu_{c_i}\|_2^2 \\ \text{subject to} & c \in \{1, \ldots, k\}^m, \end{array} \tag{8.18}$$

where the optimization variables are $c \in \mathbf{R}^m$ and $\mu_1, \ldots, \mu_k \in \mathbf{R}^n$. Obviously, minimizing the objective of (8.18) requires each data point to be assigned to the cluster representative that is closest to it, *i.e.*, an optimal cluster assignment $c^\star \in \mathbf{R}^m$ must satisfy

$$c_i^\star = \operatorname*{argmin}_{j \in \{1, \ldots, k\}} \|x_i - \mu_j^\star\|_2^2, \quad i = 1, \ldots, m,$$

where $\mu_1^\star, \ldots, \mu_k^\star \in \mathbf{R}^n$ are the corresponding optimal cluster representatives. The equivalence between the problems (8.17) and (8.18) then follows directly.

Mixture model interpretation

A simple change of variables transforms the problem (8.18) into a special case of the mixture model inverse problem of the form (8.3). In particular, we can introduce the latent factors $z_1, \ldots, z_m \in \mathbf{R}^k$, where $z_i \in \{e_1, \ldots, e_k\}$ is the standard basis vector that provides a one-hot encoding for the cluster assignment of the data point x_i, then the problem (8.18) is equivalent to

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^m z_i^T r_i \\ \text{subject to} & r_{ij} = \|x_i - \mu_j\|_2^2, \quad i = 1, \ldots, m, \quad j = 1, \ldots, k \\ & z_i \in \{0,1\}^k, \quad \mathbf{card}\, z_i = 1, \quad i = 1, \ldots, m, \end{array} \tag{8.19}$$

where the optimization variables are $z_1, \ldots, z_m \in \mathbf{R}^k$ and $\mu_1, \ldots, \mu_k \in \mathbf{R}^n$.

As a result, the clustering problem (8.18) can be interpreted, in terms of (8.19), as an inverse problem for fitting a mixture model to the dataset $x_1, \ldots, x_m$, where the base models correspond to finding the *best representative singleton* of a group of points, *i.e.*, a point that minimizes the total Euclidean distance to the points in that group (see also exercise 4.2).

8.2.2 The k-means algorithm

We can approximately solve the nonconvex problem (8.19) by the probability simplex relaxation and alternating minimization heuristic described on page 295. In particular, we can relax the problem (8.19) to the biconvex program

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^m z_i^T r_i \\ \text{subject to} & r_{ij} = \|x_i - \mu_j\|_2^2, \quad i = 1, \ldots, m, \quad j = 1, \ldots, k \\ & 0 \preceq z_i \preceq \mathbf{1}, \quad \mathbf{1}^T z_i = 1, \quad i = 1, \ldots, m, \end{array} \tag{8.20}$$

where the optimization variables are $z_1, \ldots, z_m \in \mathbf{R}^k$ and $\mu_1, \ldots, \mu_k \in \mathbf{R}^n$. Then we iterate between solving the following two convex subproblems:

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^m z_i^T \tilde{r}_i \\ \text{subject to} & 0 \preceq z_i \preceq \mathbf{1}, \quad \mathbf{1}^T z_i = 1, \quad i = 1, \ldots, m \end{array} \tag{8.21}$$

with variables $z_1, \ldots, z_m \in \mathbf{R}^k$, where

$$\tilde{r}_i = \left(\|x_i - \tilde{\mu}_1\|_2^2, \ \ldots, \ \|x_i - \tilde{\mu}_k\|_2^2 \right) \in \mathbf{R}^k, \quad i = 1, \ldots, m,$$

are problem data for fixed cluster representatives $\tilde{\mu}_1, \ldots, \tilde{\mu}_k \in \mathbf{R}^n$, and

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^m \tilde{z}_i^T r_i \\ \text{subject to} & r_{ij} = \|x_i - \mu_j\|_2^2, \quad i = 1, \ldots, m, \quad j = 1, \ldots, k \end{array} \tag{8.22}$$

with variables $\mu_1, \ldots, \mu_k \in \mathbf{R}^n$ and fixed cluster assignments $\tilde{z}_1, \ldots, \tilde{z}_m \in \mathbf{R}^k$.

The subproblems (8.21) and (8.22) both have closed-form solutions. For fixed cluster representatives $\tilde{\mu}_1, \ldots, \tilde{\mu}_k \in \mathbf{R}^n$, suppose there is no tie in the distances

$\|x_i - \tilde{\mu}_1\|_2^2, \ldots, \|x_i - \tilde{\mu}_k\|_2^2$ for $i = 1, \ldots, m$, *i.e.*, the closest cluster representative to each data point is unique with index, say,

$$j_i = \operatorname*{argmin}_{j \in \{1,\ldots,k\}} \|x_i - \tilde{\mu}_j\|_2^2, \quad i = 1, \ldots, m,$$

then the optimal point of the problem (8.21) is given by

$$z_i^\star = e_{j_i}, \quad i = 1, \ldots, m,$$

where $e_1, \ldots, e_k \in \mathbf{R}^k$ are the standard basis vectors in $\mathbf{R}^k$ (see also remark 8.2). On the other hand, when the cluster assignments $\tilde{z}_1, \ldots, \tilde{z}_m \in \mathbf{R}^k$ are fixed, the optimal point of the problem (8.22) is given by

$$\mu_j^\star = \frac{\sum_{i=1}^m \tilde{z}_{ij} x_i}{\sum_{i=1}^m \tilde{z}_{ij}}, \quad j = 1, \ldots, k,$$

which is a convex combination of the data points $x_1, \ldots, x_m$ for $j = 1, \ldots, k$, with coefficients proportional to the cluster assignments $\tilde{z}_{1j}, \ldots, \tilde{z}_{mj}$. Note that when $\tilde{z}_1, \ldots, \tilde{z}_m \in \{e_1, \ldots, e_k\}$, *i.e.*, each data point is assigned to exactly one group, the optimal cluster representatives $\mu_1^\star, \ldots, \mu_k^\star$ given above are simply the means of the data points in each group (*cf.* exercise 4.3).

Applying the ideas presented above directly to the original clustering problem (8.18) leads to the well-known *k-means algorithm*, which is summarized below.

Algorithm 8.1 k-MEANS ALGORITHM.

given the dataset $x_1, \ldots, x_m \in \mathbf{R}^n$ and an initial choice of cluster representatives $\tilde{\mu}_1, \ldots, \tilde{\mu}_k \in \mathbf{R}^n$.

repeat

1. *Cluster assignment.* Assign each data point to the nearest cluster representative:
$$\tilde{c}_i := \operatorname*{argmin}_{j \in \{1,\ldots,k\}} \|x_i - \tilde{\mu}_j\|_2^2, \quad i = 1, \ldots, m.$$
2. *Group formation.* Form the groups based on the cluster assignments:
$$G_j := \{i \in \{1, \ldots, m\} \mid \tilde{c}_i = j\}, \quad j = 1, \ldots, k.$$
3. *Representative update.* Set cluster representatives to be the mean of the data points in each group:
$$\tilde{\mu}_j := (1/|G_j|) \sum_{i \in G_j} x_i, \quad j = 1, \ldots, k.$$

until convergence.

Since algorithm 8.1 for approximately solving the problem (8.18) is equivalent to applying the alternative convex search heuristic to the biconvex program (8.20), it is guaranteed that after each iteration, the objective of (8.18) is nonincreasing and hence converges.

Figure 8.5 shows several example iterations of the k-means algorithm 8.1 applied to a randomly generated dataset. Figure 8.6 shows the final clustering result on this dataset and the convergence path of the cluster representatives $\tilde{\mu}_1, \ldots, \tilde{\mu}_k \in \mathbf{R}^n$ over iterations.

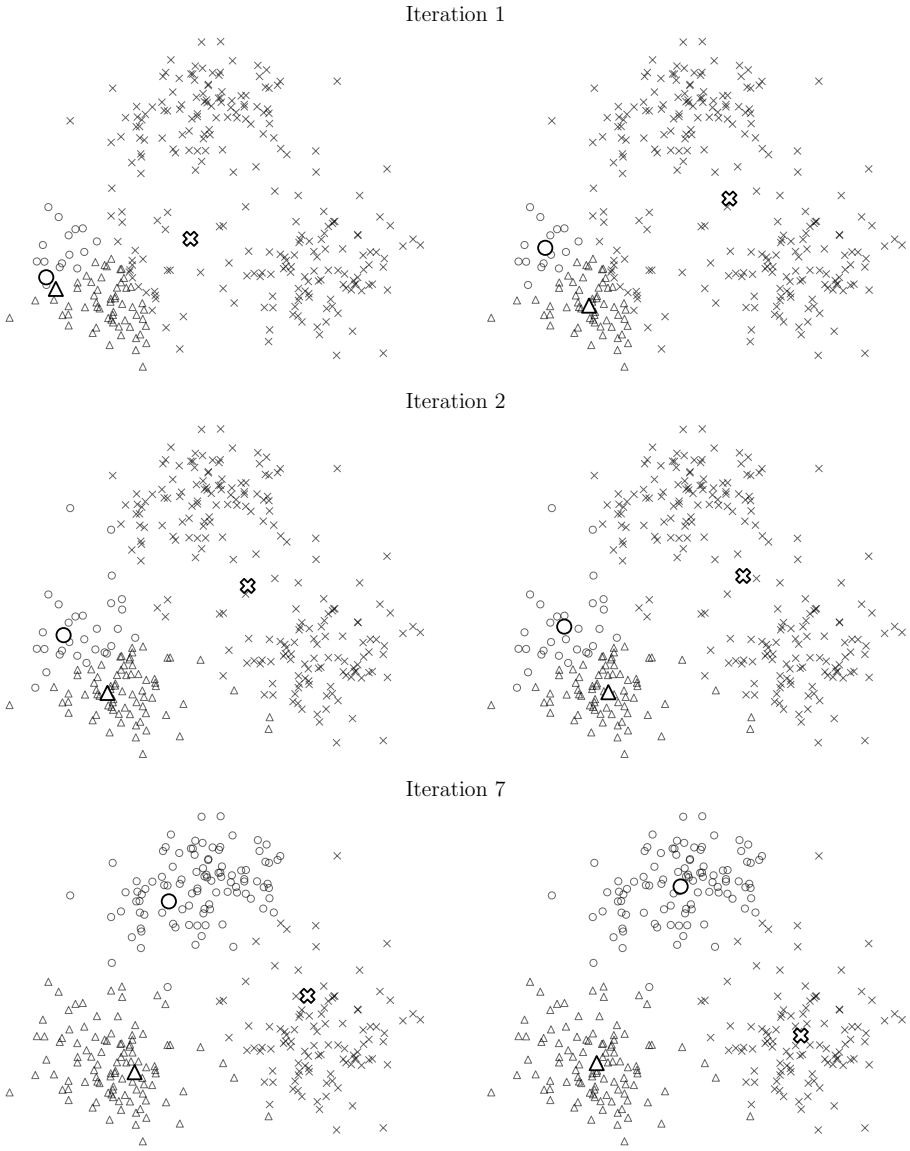

Figure 8.5 Three example iterations of the k-means algorithm. The cluster representatives are shown thicker. In each row, the left plot shows the cluster assignment step (algorithm 8.1, step 1), and the right plot shows the updated cluster representatives according to step 3 of the algorithm.

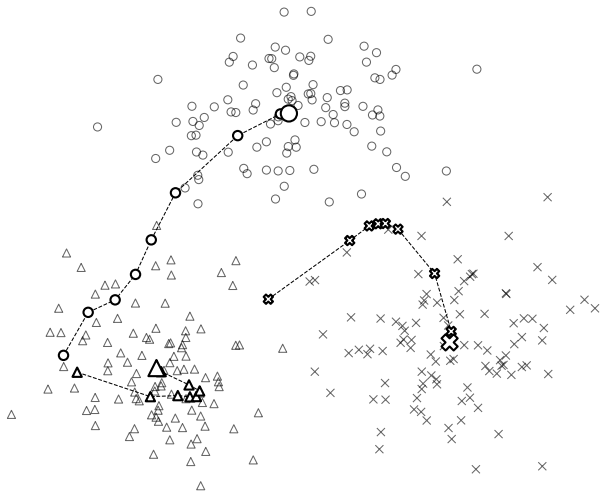

Figure 8.6 Final clustering result corresponding to figure 8.5. The cluster representatives are shown thicker, and the final representatives are shown larger. The convergence path of the cluster representatives over iterations is shown as dashed curves.

8.2.3 Clustering with prior information

Representing a clustering problem in terms of the biconvex program (8.20) is particularly useful when we want to add some regularization terms or constraints to incorporate some prior knowledge about the cluster representatives $\mu_1, \ldots, \mu_k \in \mathbf{R}^n$ or the cluster assignments $z_1, \ldots, z_m \in \mathbf{R}^k$. In these cases, there might not be closed-form solutions for the subproblems (8.21) and (8.22), but they are still convex optimization problems that can be solved efficiently (provided the regularization terms and constraints are convex).

We will give some examples of incorporating prior information into clustering problems as follows. Here we restrict our discussion to constraints, although many of them can also be turned into regularization terms if properly translated (see, *e.g.*, §6.1.1, page 196).

Assignment and relation constraints

We could add some constraints on the cluster assignments of the data points, which can be useful when we have some prior information about the relationships between the data points. Suppose we know that the data point x_i is only allowed to belong to one of a specific subset of clusters, say, $S_i \subseteq \{1, \ldots, k\}$, then we can add the following *assignment constraint* to the problem (8.20):

$$z_{ij} = 0, \quad j \in \{1, \ldots, k\} \setminus S_i,$$

where z_{ij} is the jth component of the cluster assignment variable $z_i \in \mathbf{R}^k$ in (8.20). As an extreme case, if we know that the data point x_i must be assigned to the jth cluster for some $j \in \{1, \ldots, k\}$, then we have the constraint

$$z_{ij} = 1.$$

These constraints can be expressed as a finite number of linear equality constraints in the latent factor variables $z_1, \ldots, z_m \in \mathbf{R}^k$ of the problem (8.20).

We could also consider the pairwise relationships between the data points x_p and x_q for $p, q \in \{1, \ldots, m\}$. For example, if we know that the data points x_p and x_q must belong to the same cluster, then we can add the following *must-link constraint* to the problem (8.20):

$$z_p = z_q,$$

or equivalently,

$$z_{pj} = z_{qj}, \quad j = 1, \ldots, k, \tag{8.23}$$

which are all linear equality constraints. Conversely, if we know that the data points x_p and x_q must belong to different clusters, then we have the *cannot-link constraint* given by

$$z_{pj} + z_{qj} \leq 1, \quad j = 1, \ldots, k. \tag{8.24}$$

When $z_p, z_q \in \{e_1, \ldots, e_k\}$, the constraint (8.24) requires that for each $j = 1, \ldots, k$, only one of z_{pj} and z_{qj} can be equal to 1, which therefore ensures that the data points x_p and x_q are not assigned to the same cluster. However, we should carefully interpret the cannot-link constraint (8.24) when $z_p, z_q \in \mathbf{R}^k$ just stay in the probability simplex in $\mathbf{R}^k$ as in the problem (8.20). In this case, the constraint (8.24) says that, for each $j = 1, \ldots, k$, if z_{pj} is close to 1, *i.e.*, the data point x_p is likely to be assigned to the jth cluster, then z_{qj} must be close to 0, so the data point x_q is unlikely to be assigned to the same cluster. In other words, the data points x_p and x_q cannot be assigned to the same cluster both with high probability. Similarly, from a probabilistic perspective, the must-link constraint (8.23) says that the data points x_p and x_q must be assigned to the same cluster with equal probability.

Size constraints

We could add some constraints on the sizes of the clusters, *i.e.*, to control the number of data points assigned to each group. For example, let

$$s_j = \sum_{i=1}^{m} z_{ij}, \quad j = 1, \ldots, k,$$

be a size measurement of the jth cluster for $j = 1, \ldots, k$, where z_{ij} is the jth component of the cluster assignment variable $z_i \in \mathbf{R}^k$ for $i = 1, \ldots, m$, according to the problem (8.20). Note that when $z_1, \ldots, z_m \in \{e_1, \ldots, e_k\}$, the size measurement s_j becomes an integer and is exactly the number of data points assigned to the jth cluster. The *size constraint*

$$L_j \leq s_j \leq U_j, \quad j = 1, \ldots, k, \tag{8.25}$$

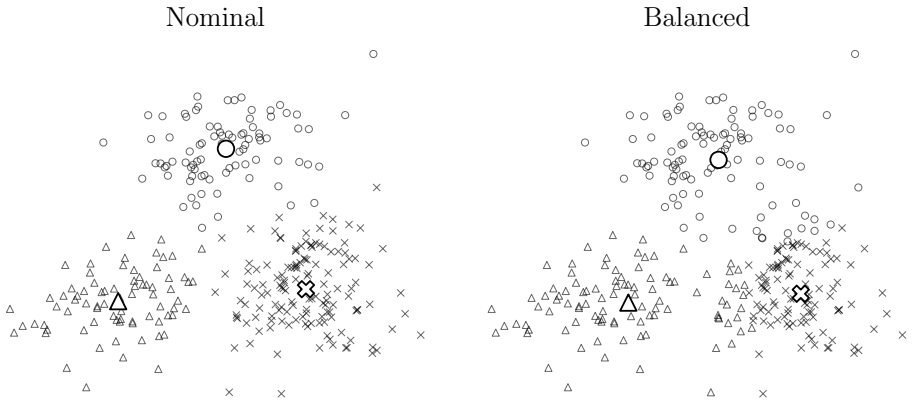

Figure 8.7 The nominal clustering result (*left*) from the problem (8.20), and the balanced clustering result (*right*) under the constraint (8.26), on the same dataset. The final cluster representatives are shown thicker.

with given lower and upper bounds $L_j, U_j \in \mathbf{R}$ for $j = 1, \ldots, k$, require that the size of the jth cluster must be between L_j and U_j. A special case of this constraint is the *balance constraint*, which is given by

$$\left| s_j - \frac{m}{k} \right| \leq \epsilon, \quad j = 1, \ldots, k, \tag{8.26}$$

for some small given $\epsilon > 0$. This constraint requires that the size of each cluster must be close to m/k, which hence ensures the number of points in all clusters should approximately be the same. We could express the balance constraint (8.26) in the form of (8.25) as

$$\frac{m}{k} - \epsilon \leq s_j \leq \frac{m}{k} + \epsilon, \quad j = 1, \ldots, k.$$

Example 8.3 *Balanced clustering.* We compare the clustering results between the nominal clustering problem (8.20), and the problem with the additional balance constraint (8.26), on a random generated dataset with $m = 300$ data points. The dataset was generated so that the size of the top and bottom left groups is around 80, while the size of the bottom right group is around 140. Our goal is to cluster these data points into $k = 3$ groups, and we set $\epsilon = 3$ in the balance constraint (8.26), which requires that the size of each cluster should be between 97 and 103.

Figure 8.7 shows a set of clustering results from solving these two problems via alternate convex search, where the nominal clustering results are shown in the left plot, and the balanced clustering results are shown in the right plot. We have several observations from the figure:

- The nominal clustering result is more consistent with the natural (*i.e.*, visual) grouping of the data points, while the balanced clustering result is more distorted to the right.
- The balanced clustering assigns more data points distributed in the middle of the dataset to the top and left clusters, so that the sizes of the three clusters are

more balanced than those in the nominal clustering. As a result, the sizes of the three clusters from the nominal clustering are 84, 77, and 139, while the sizes of the three clusters from the balanced clustering are 98, 99, and 103.

A simple generalization of the size constraint (8.25) is the *capacity constraint*, which is given by

$$C_j^{\min} \leq \sum_{i=1}^{m} w_i z_{ij} \leq C_j^{\max}, \quad j = 1, \ldots, k,$$

where $w \in \mathbf{R}^m$ is a given weight vector for the data points, with the ith component representing the cost of assigning data point x_i to any cluster, and $C_j^{\min}, C_j^{\max} \in \mathbf{R}$ are given lower and upper bounds for the total cost for the jth cluster. These constraints are all linear inequality constraints in the latent factor variables $z_1, \ldots, z_m \in \mathbf{R}^k$ of the problem (8.20).

As a more complex example, suppose the data points are partitioned into q categories $G_1, \ldots, G_q \subseteq \{1, \ldots, m\}$ with

$$\bigcup_{i=1}^{q} G_i = \{1, \ldots, m\} \quad \text{and} \quad G_i \cap G_j = \emptyset \text{ for } i \neq j,$$

and we want to control the number of data points assigned to each cluster in each category. If the category G_p should make up between L_p and U_p fraction of each cluster for $p = 1, \ldots, q$, then we can add the following constraints to the problem (8.20):

$$L_p s_j \leq \sum_{i \in G_p} z_{ij} \leq U_p s_j, \quad p = 1, \ldots, q, \quad j = 1, \ldots, k,$$

where $L_p, U_p \in [0, 1]$ are given for $p = 1, \ldots, q$. Substituting $s_j = \sum_{i=1}^{m} z_{ij}$ into the above constraints, we can express them as

$$L_p \sum_{i=1}^{m} z_{ij} \leq \sum_{i \in G_p} z_{ij} \leq U_p \sum_{i=1}^{m} z_{ij}, \quad p = 1, \ldots, q, \quad j = 1, \ldots, k,$$

which are linear inequality constraints in $z_1, \ldots, z_m \in \mathbf{R}^k$. Similarly, if we want the proportion of the data points in category G_p in each cluster to be approximately $\eta_p \in [0, 1]$ for $p = 1, \ldots, q$, then we have the following constraints:

$$\left| \sum_{i \in G_p} z_{ij} - \eta_p \sum_{i=1}^{m} z_{ij} \right| \leq \epsilon, \quad p = 1, \ldots, q, \quad j = 1, \ldots, k,$$

where $\epsilon > 0$ is a given acceptance threshold.

Geometric constraints

When the data points $x_1, \ldots, x_m \in \mathbf{R}^n$ have some geometric meaning, *e.g.*, they are locations of some objects in a physical space, we might want to add some *geometric*

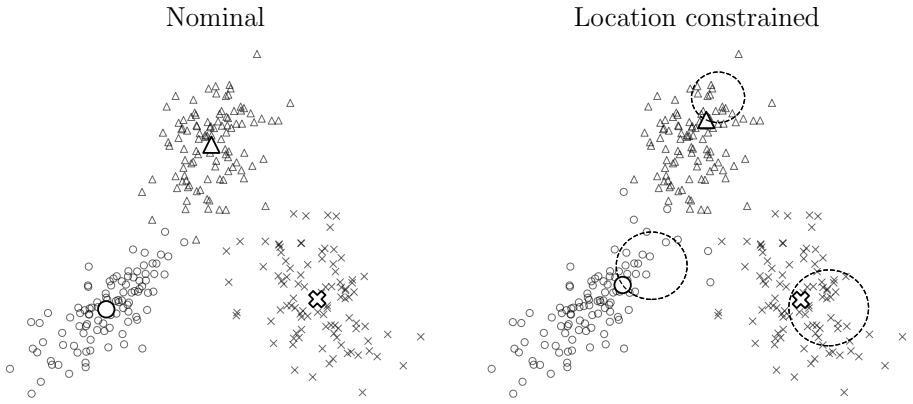

Figure 8.8 The nominal clustering result (*left*) from the problem (8.20), and the geometrically constrained clustering result (*right*) under the constraint (8.27), on the same dataset. The final cluster representatives are shown thicker. The dashed circles in the right plot show the location constraints for the cluster representatives.

constraints on the cluster representatives $\mu_1, \ldots, \mu_k \in \mathbf{R}^n$ to the problem (8.20). For example, the linear inequality constraint

$$A_i \mu_i \preceq b_i, \quad i = 1, \ldots, k,$$

where $A_i \in \mathbf{R}^{p_i \times n}$ and $b_i \in \mathbf{R}^{p_i}$ are given for $i = 1, \ldots, k$, requires that the cluster representative μ_i must lie in the polyhedron $\{u \in \mathbf{R}^n \mid A_i u \preceq b_i\}$. Similarly, if μ_i should lie in the ball $\{u \in \mathbf{R}^n \mid \|u - c_i\|_2 \leq r_i\}$ for some given center $c_i \in \mathbf{R}^n$ and radius $r_i > 0$, then we can add the following convex constraint to the problem (8.20):

$$\|\mu_i - c_i\|_2 \leq r_i, \quad i = 1, \ldots, k. \tag{8.27}$$

These ideas are readily adapted to other geometric shapes such as ellipsoids.

Example 8.4 *Clustering with location constraints.* We compare the nominal and geometrically constrained clustering on a randomly generated dataset with $m = 300$ data points. In particular, we cluster these data points into $k = 3$ groups, via the problem (8.20) with and without the additional location constraint (8.27), solved by alternate convex search.

Figure 8.8 shows a set of clustering results from these two problems, where the nominal clustering outcome is shown in the left plot, and the geometrically constrained clustering outcome is shown in the right plot. The dashed circles in the right plot draw the location constraints for the cluster representatives under (8.27), *i.e.*, the cluster representatives must lie in the corresponding circles. From the figure, we can see that the location constraints significantly distort the clustering result. In particular, the cluster representatives of the top and left clusters are located at the boundary of the corresponding circles, which indicates that the location constraints are active for these two clusters.

Geometric constraints might also involve the data points $x_1, \ldots, x_m \in \mathbf{R}^n$. As an example, we might want the maximum distance between any data point in the jth cluster and the corresponding cluster representative μ_j to be at most R_j. In other words, we want to ensure that the smallest ball centered at μ_j that contains all the data points in the jth cluster has radius at most R_j for $j = 1, \ldots, k$. This constraint can be expressed as

$$\max_{i \in G_j} \|x_i - \mu_j\|_2 \leq R_j, \quad j = 1, \ldots, k, \tag{8.28}$$

with

$$G_j = \left\{ i \in \{1, \ldots, m\} \;\middle|\; \operatorname*{argmax}_{q=1,\ldots,k} z_{iq} = j \right\}, \quad j = 1, \ldots, k, \tag{8.29}$$

where $z_1, \ldots, z_m \in \mathbf{R}^k$ are the cluster assignment variables in the problem (8.20). However, the constraint given by (8.28) and (8.29) is nonconvex. One possible convex relaxation of this constraint is given by

$$\|x_i - \mu_j\|_2 \leq R_j + M(1 - z_{ij}), \quad i = 1, \ldots, m, \quad j = 1, \ldots, k, \tag{8.30}$$

where $M > 0$ is a sufficiently large constant, *e.g.*, the diameter of the smallest ball that contains all the data points $x_1, \ldots, x_m \in \mathbf{R}^n$. For each $i = 1, \ldots, m$ and $j = 1, \ldots, k$, if z_{ij} is close to 1, then the constraint (8.30) is approximately equivalent to $\|x_i - \mu_j\|_2 \leq R_j$, which requires that the data point x_i must be within the ball centered at μ_j with radius R_j. On the other hand, if z_{ij} is close to 0, then the above constraint is approximately $\|x_i - \mu_j\|_2 \leq R_j + M$, which is always satisfied when M is sufficiently large. Therefore, the relaxation (8.30) requires that if the data point x_i is assigned to the jth cluster with high probability, then it must be roughly within the ball centered at μ_j with radius R_j.

8.3 Principal component analysis

This section deals with approximation problems involving a data matrix of the form

$$A = \begin{bmatrix} a_1^T \\ \vdots \\ a_m^T \end{bmatrix} \in \mathbf{R}^{m \times n},$$

where $a_1^T, \ldots, a_m^T$ are the *row* vectors of A. In practice, the vector $a_i \in \mathbf{R}^n$ is often interpreted as the *feature vector* of the ith data point for $i = 1, \ldots, m$, where its jth component $a_{ij} \in \mathbf{R}$ represents the jth feature value of the data point. In this context, the jth column of A is then interpreted as the vector of the jth feature across all data points.

In practice, it is commonly assumed that the number of data points m is much larger than the number of features n, *i.e.*, $m \gg n$, so that the data matrix A is tall. In the following discussion, we will additionally assume the data matrix $A \in \mathbf{R}^{m \times n}$ is *standardized*, *i.e.*, each column of A has zero mean and unit variance, for the

reason that will become clear later. In practice, we can always standardize a given data matrix A by the following linear transformation:

$$\tilde{A} = (A - \mathbf{1}\mu^T)D^{-1}, \tag{8.31}$$

where $\mu \in \mathbf{R}^n$ is the mean vector of the columns of A, *i.e.*,

$$\mu_j = \frac{1}{m}\sum_{i=1}^{m} a_{ij}, \quad j = 1, \ldots, n,$$

and

$$D = \mathbf{diag}(\sigma_1, \ldots, \sigma_n) \in \mathbf{R}^{n\times n}$$

is a diagonal matrix with $\sigma_j > 0$ being the standard deviation of the jth column of A, *i.e.*,

$$\sigma_j = \left(\frac{1}{m}\sum_{i=1}^{m}(a_{ij} - \mu_j)^2\right)^{1/2}, \quad j = 1, \ldots, n.$$

The linear transformation (8.31) is sometimes called *standardization* or *z-score normalization* of the feature matrix A. The resulting matrix $\tilde{A} \in \mathbf{R}^{m\times n}$ from (8.31) has the same shape as A and is sometimes called the *standardized feature matrix*.

8.3.1 Low rank approximation

The problem of *principal component analysis* (PCA) is to find the best *low rank approximation* $Z \in \mathbf{R}^{m\times n}$ of the data matrix $A \in \mathbf{R}^{m\times n}$ that minimizes the approximation error, in terms of the Frobenius norm, *i.e.*, solves the problem

$$\begin{array}{ll} \text{minimize} & \|A - Z\|_F^2 \\ \text{subject to} & \mathbf{rank}\, Z \leq k, \end{array} \tag{8.32}$$

where the optimization variable is $Z \in \mathbf{R}^{m\times n}$. The positive integer $k < n$ is a given problem parameter that specifies the maximum acceptable rank of the approximation matrix Z. Expanding this objective function according to the definition of the Frobenius norm, we have

$$\|A - Z\|_F^2 = \sum_{i=1}^{m}\|a_i - z_i\|_2^2 = \sum_{i=1}^{m}\sum_{j=1}^{n}(a_{ij} - z_{ij})^2,$$

where $a_1^T, \ldots, a_m^T$ and $z_1^T, \ldots, z_m^T$ are the row vectors of the matrices A and Z, respectively. Therefore, a direct interpretation of the problem (8.32) of low rank approximation is to find a matrix $Z \in \mathbf{R}^{m\times n}$ with rank at most k such that the sum of the squared distances between the corresponding row vectors of A and Z is minimized. Equivalently, it seeks to minimize the quadratic error between the corresponding entries of A and Z.

We could directly encode the rank-k constraint in (8.32) by factorizing the variable Z into the product of two matrices $X \in \mathbf{R}^{m\times k}$ and $Y \in \mathbf{R}^{k\times n}$, so that $Z = XY$.

(We will see later that, this factorization is, of course, not unique.) Then the low rank approximation problem (8.32) can be equivalently written as

$$\text{minimize} \quad \|A - XY\|_F^2 \tag{8.33}$$

with variables $X \in \mathbf{R}^{m\times k}$ and $Y \in \mathbf{R}^{k\times n}$.

Solution via singular value decomposition

It is easily seen that the problem (8.32) is nonconvex, because of the the rank constraint. The equivalent problem (8.33) is also not convex, but it is biconvex in the variables X and Y, since when one of them is fixed, the problem becomes a convex quadratic program in the other variable. Even though both problems are not convex optimization problems, they can, in fact, be solved (globally) in closed-form via the *singular value decomposition* (SVD) of the data matrix A.

Assuming that the data matrix $A \in \mathbf{R}^{m\times n}$ has $\mathbf{rank}\, A = r$, we have the (compact) SVD of A given by

$$A = U\Sigma V^T,$$

where $U \in \mathbf{R}^{m\times r}$ and $V \in \mathbf{R}^{n\times r}$ are matrices with orthonormal columns, and

$$\Sigma = \mathbf{diag}(\sigma_1, \ldots, \sigma_r) \in \mathbf{R}^{r\times r}$$

is a diagonal matrix with $\sigma_1 \geq \cdots \geq \sigma_r > 0$ being the *singular values* of A (see §A.2.2). Then the best rank-k ($k \leq r$) approximation $Z^\star \in \mathbf{R}^{m\times n}$ of A that solves the problem (8.32) is given by

$$Z^\star = U_k\Sigma_k V_k^T, \tag{8.34}$$

where $U_k \in \mathbf{R}^{m\times k}$ and $V_k \in \mathbf{R}^{n\times k}$ are the matrices consisting of the first k columns of U and V, respectively, and $\Sigma_k = \mathbf{diag}(\sigma_1, \ldots, \sigma_k) \in \mathbf{R}^{k\times k}$ is the diagonal matrix consisting of the first k singular values of A. Correspondingly, a factorization of the best rank-k approximation $Z^\star = X^\star Y^\star$ that solves the problem (8.33) is given by

$$X^\star = U_k\Sigma_k^{1/2} \in \mathbf{R}^{m\times k} \quad \text{and} \quad Y^\star = \Sigma_k^{1/2} V_k^T \in \mathbf{R}^{k\times n}. \tag{8.35}$$

It is a famous result in linear algebra that the SVD solution given by (8.34) (and (8.35)) is the best rank-k approximation of A that minimizes $\|A - Z\|_F^2$ over all matrices $Z \in \mathbf{R}^{m\times n}$ with $\mathbf{rank}\, Z \leq k$. Although it is not very difficult to show this result (at least semi-intuitively), we will not give the proof here but refer the interested readers to the references.

Remark 8.3 *Uniqueness of the SVD solution.* We should note that solutions to the factorized problem (8.33) are not unique. Since for an optimal point $(X^\star, Y^\star)$ given by (8.35), suppose $W \in \mathbf{R}^{k\times k}$ is any invertible matrix, then we have

$$(X^\star W)(W^{-1}Y^\star) = X^\star Y^\star = Z^\star, \tag{8.36}$$

which implies that the matrices $X^\star W$ and $W^{-1}Y^\star$ provide another factorization of the same optimal matrix $Z^\star$ to the problem (8.32).

As a special case, we could take the scaling factor in (8.36) to be a positive scalar $c > 0$, then the scaled optimal matrices given by

$$cX^\star \in \mathbf{R}^{m\times k} \quad \text{and} \quad c^{-1}Y^\star \in \mathbf{R}^{k\times n}$$

are also optimal for the problem (8.33). From this, we could see that the set of optimal points to the problem (8.33) is even unbounded.

8.3.2 Interpretations

We could give the problem of low rank approximation (8.32) (as well as the factorized problem (8.33)) several interpretations from different perspectives.

Geometric interpretation

We could interpret each row a_i^T of the data matrix $A \in \mathbf{R}^{m\times n}$ as a data point in $\mathbf{R}^n$ for $i = 1, \ldots, m$. Notice that requiring the approximation matrix $Z \in \mathbf{R}^{m\times n}$ to have rank at most k is equivalent to requiring that the row vectors of Z must lie in a k-dimensional subspace of $\mathbf{R}^n$. Therefore, we can interpret the problem (8.32) as finding a k-dimensional subspace in $\mathbf{R}^n$ such that the sum of the squared distances between the data points $a_1, \ldots, a_m \in \mathbf{R}^n$ and their projections $z_1, \ldots, z_m \in \mathbf{R}^n$ (whose transposes are the rows of Z) onto this subspace is minimized. The following example illustrates this idea.

Example 8.5 *Low rank approximation.* Consider a data matrix $A \in \mathbf{R}^{m\times n}$ with $m = 600$ and $n = 2$, where each data point, *i.e.*, each row of A, is generated from a zero mean multivariate Gaussian distribution. We plot the data points in $\mathbf{R}^2$ in figure 8.9 (shown crosses), from which we see that the data points are distributed roughly in an flat elliptical shape. This suggests that the data points are approximately distributed along a line (through the origin) in $\mathbf{R}^2$.

We could apply the low rank approximation problem (8.32) to this dataset, with $k = 1$, to find the best rank-1 approximation of the data matrix A. This is equivalent to finding a one-dimensional subspace in $\mathbf{R}^2$ (which is a line through the origin) that best approximates the given data points, in the sense that the sum of the squared distances between the data points and their corresponding projections onto the subspace is minimized. This subspace obtained from solving the problem (8.32) is shown as the dashed line in figure 8.9, and the circles indicate the projections of the data points onto the subspace. We can see that the approximation captures the main direction of the data points, *i.e.*, the major axis of the elliptical distribution, along which the data points have the largest variation.

This geometric interpretation of the low rank approximation problem (8.32) is universal no matter what data matrix $A \in \mathbf{R}^{m\times n}$ we have. However, in practice, we should be careful when the data matrix A is not standardized (in particular, not centered). In this case, the data points $a_1, \ldots, a_m \in \mathbf{R}^n$ might not be centered around the origin, but the row vectors of the approximation matrix $Z \in \mathbf{R}^{m\times n}$ are still required to lie in a k-dimensional subspace of $\mathbf{R}^n$, which contains the origin. As a result, the approximation from (8.32) might be very poor. Numerically, this

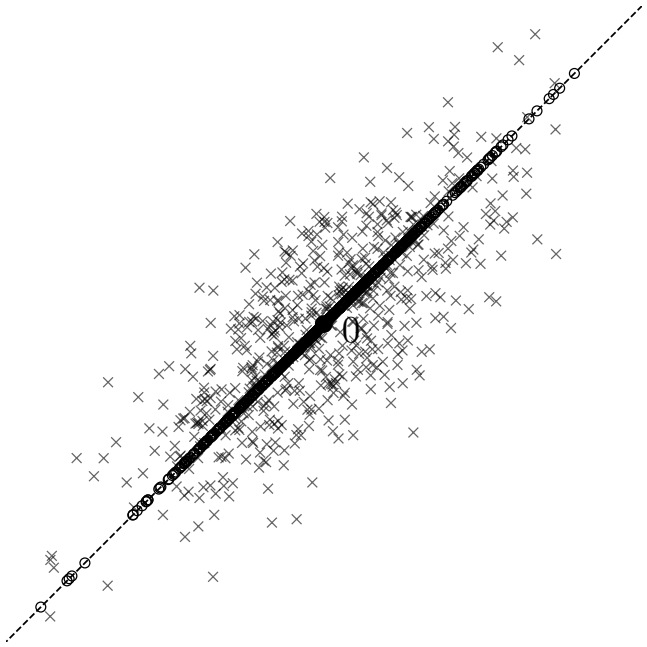

Figure 8.9 Plot of 600 data points in $\mathbf{R}^2$ generated from a zero mean multivariate Gaussian distribution, shown as crosses, which correspond to the rows of a data matrix $A \in \mathbf{R}^{600\times 2}$. The circles are the rows of the best rank-1 approximation $Z^\star \in \mathbf{R}^{600\times 2}$ of A from solving the problem (8.32) with $k = 1$, which lies in a line through the origin (shown dashed).

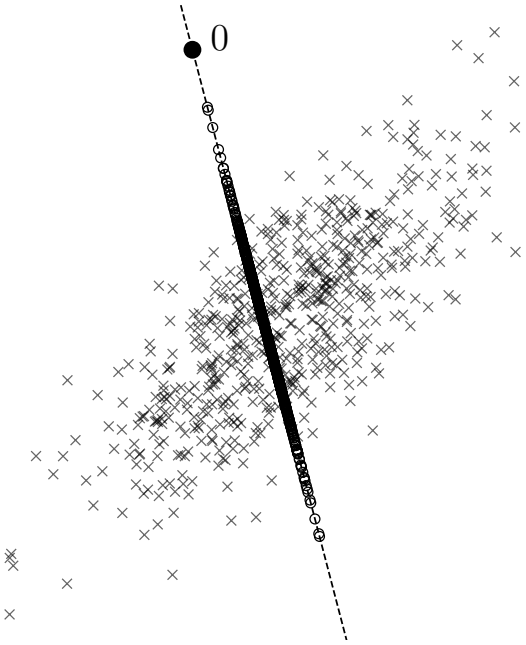

Figure 8.10 Plot of the same data points (shown crosses) as in figure 8.9, but translated away from the origin. The dashed line shows the subspace corresponding to the best rank-1 approximation of the dataset from solving (8.32), and the circles indicate the projections of the data points onto this subspace. Here the origin (shown dot) is not at the center of the data points.

means that the optimal value of the problem (8.32) might be very large, compared to the case when the data matrix A is centered. Practically, this means that the low rank approximation from (8.32) might not capture the major variation direction of the data points, which is often our main interest in applications. (We will see later that this result can be obtained more formally from a statistical point of view.)

Figure 8.10 shows an example of this phenomenon, where we solve the problem (8.32) with $k = 1$ on the same dataset as in figure 8.9, but translated away from the origin by adding some constant to the data points. For this nonstandardized dataset, the one-dimensional subspace (*i.e.*, the line) corresponding to the solution of (8.32) is no longer aligned with the major axis of the data distribution ellipse, although the projections of the data points onto this line still gives the best rank-1 approximation of the data points in terms of minimizing the sum of squared distances.

For this reason, our initial assumption that the data matrix A is standardized now makes more sense.

Statistical interpretation

The low rank approximation problem (8.32), or the idea of principal component analysis, is traditionally interpreted from a statistical point of view. Suppose the data points $a_1, \ldots, a_m \in \mathbf{R}^n$ whose transposes form the rows of the data matrix $A \in \mathbf{R}^{m\times n}$ are instances of a random vector from some probability distribution in $\mathbf{R}^n$ with zero mean (*i.e.*, A is columnwise centered). Statistically, the low rank approximation problem (8.32) can be interpreted as finding k orthogonal directions in $\mathbf{R}^n$ such that the projection of the data points onto these directions has the top k largest variance.

Remark 8.4 *Proof of the statistical interpretation.* We show this result for a simple case where $k = 1$, and leave the generalization to $k > 1$ as an exercise (exercise 8.3). Let $v \in \mathbf{R}^n$ be a unit vector that represents the direction of some one-dimensional subspace in $\mathbf{R}^n$, then

$$a_1^T v, \ldots, a_m^T v \in \mathbf{R}$$

are the coefficients of the projection of the data points $a_i \in \mathbf{R}^n$ onto this direction for $i = 1, \ldots, m$. Assuming that the data points have zero mean, the empirical variance of the projection coefficients $a_1^T v, \ldots, a_m^T v$ is given by

$$\frac{1}{m}\sum_{i=1}^{m}(a_i^T v)^2 = \frac{1}{m}\|Av\|_2^2 = \frac{1}{m}v^T A^T A v. \tag{8.37}$$

Therefore, finding the direction where the variance of the projection coefficients is maximized is equivalent to solving the following optimization problem:

$$\begin{array}{ll} \text{maximize} & v^T A^T A v \\ \text{subject to} & \|v\|_2 = 1, \end{array} \tag{8.38}$$

where the optimization variable is $v \in \mathbf{R}^n$.

Now since the matrix $A^T A \in \mathbf{S}_+^n$ is symmetric positive semidefinite, it has an *eigenvalue decomposition* given by

$$A^T A = Q\Lambda Q^T, \tag{8.39}$$

where $Q \in \mathbf{R}^{n\times n}$ is an orthogonal matrix of the *eigenvectors* of A^TA and $\Lambda = \mathbf{diag}(\lambda_1, \ldots, \lambda_n) \in \mathbf{R}^{n\times n}$ is a diagonal matrix with $\lambda_1 \geq \cdots \geq \lambda_n \geq 0$ being the *eigenvalues* of A^TA (see §A.2.1). Since the matrix $Q \in \mathbf{R}^{n\times n}$ is orthogonal, any unit vector $v \in \mathbf{R}^n$ can be written as

$$v = Qw$$

for some vector $w \in \mathbf{R}^n$ with $\|w\|_2 = 1$. As a result, we have

$$v^TA^TAv = w^TQ^TQ\Lambda Q^TQw = w^T\Lambda w = \sum_{i=1}^n \lambda_i w_i^2.$$

Since $\sum_{i=1}^n w_i^2 = 1$, we have

$$v^TA^TAv = \sum_{i=1}^n \lambda_i w_i^2 \leq \lambda_1 \sum_{i=1}^n w_i^2 = \lambda_1,$$

which provides an upper bound on the objective of (8.38), *i.e.*, the empirical variance of the projection coefficients $a_1^Tv, \ldots, a_m^Tv \in \mathbf{R}$ (scaled by m). Obviously, this upper bound can be achieved by choosing $w = e_1$, where $e_1 \in \mathbf{R}^n$ is the first standard basis vector in $\mathbf{R}^n$. In other words, the direction $v \in \mathbf{R}^n$ that solves the problem (8.38), *i.e.*, maximizes the variance of the projection coefficients, is given by the first column of the matrix $Q \in \mathbf{R}^{n\times n}$, *i.e.*, $v = Qe_1$, which is the eigenvector of A^TA associated with its largest eigenvalue λ_1.

On the other hand, if

$$A = U\Sigma V^T$$

with $U \in \mathbf{R}^{m\times m}$, $\Sigma \in \mathbf{R}^{m\times n}$, and $V \in \mathbf{R}^{n\times n}$ is a (full) SVD of A, then we have

$$A^TA = V\Sigma^TU^TU\Sigma V^T = V\Sigma^T\Sigma V^T,$$

i.e., the matrix $V \in \mathbf{R}^{n\times n}$ of the *right singular vectors* of A can be taken as the matrix Q of eigenvectors of A^TA in (8.39), and the squares of the singular values of A are the eigenvalues of A^TA. Put together, we conclude that the direction $v \in \mathbf{R}^n$ given by the first right singular vector $v_1 = Ve_1$ of A maximizes the variance of the projection coefficients $a_1^Tv, \ldots, a_m^Tv$.

Recall that the best rank-1 approximation of A according to the problem (8.32) is given by

$$Z^\star = u_1\sigma_1v_1^T = Av_1v_1^T, \tag{8.40}$$

where $u_1 \in \mathbf{R}^m$ and $v_1 \in \mathbf{R}^n$ are the first columns of the matrices U and V, respectively, and σ_1 is the largest singular value of A. The second equality in (8.40) follows from the fact that since V is orthogonal, we have $V^Tv_1 = e_1$, and thus

$$Av_1 = U\Sigma V^Tv_1 = U\Sigma e_1 = \sigma_1u_1.$$

Rewriting the right-hand side of (8.40) in terms of the rows of A, we have

$$Z^\star = \begin{bmatrix} (a_1^Tv_1)v_1^T \\ \vdots \\ (a_m^Tv_1)v_1^T \end{bmatrix}.$$

This means that the optimal point $Z^\star \in \mathbf{R}^{m\times n}$ of the problem (8.32) with $k = 1$ is given by the projection of the data points $a_1, \ldots, a_m \in \mathbf{R}^n$ onto the direction $v_1 \in \mathbf{R}^n$ of the first right singular vector of A, which is shown to be the direction that maximizes the variance of the projection coefficients.

When the data matrix is centered at the origin, this statistical interpretation of the low rank approximation problem (8.32) is consistent with our observations in figure 8.9. On the other hand, when the data matrix A is not centered, according to remark 8.4, we see again that the low rank approximation from solving the problem (8.32) does not necessarily find the directions of the major variation of the data points. In this case, for instance, the best rank-1 approximation of A still projects the data points onto the direction that solves the problem (8.38), however, the variance of the projection coefficients is no longer given by (8.37).

This interpretation also answers the question of why in practice we usually require each column of the data matrix A to have unit variance, in addition to being centered. If the columns of A are not standardized, then the low rank approximation obtained by solving problem (8.32) may project the data points primarily along directions dominated by features with large variance. However, such dominance might simply reflect differences in measurement units or scales, rather than more meaningful patterns of variation among the data points.

Maximum likelihood estimation

We consider the low rank approximation problem (8.33) in factorized form from a probabilistic modeling perspective. Suppose each entry $a_{ij} \in \mathbf{R}$ of the data matrix $A \in \mathbf{R}^{m\times n}$ is generated from a linear measurement model given by

$$a_{ij} = x_i^T y_j + \epsilon_{ij}, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n,$$

where $x_i \in \mathbf{R}^k$ and $y_j \in \mathbf{R}^k$ are the coefficient vectors associated with the ith row and the jth column of A, respectively, and $\epsilon_{ij} \in \mathbf{R}$ are IID Gaussian noise with zero mean. Let

$$X = \begin{bmatrix} x_1^T \\ \vdots \\ x_m^T \end{bmatrix} \in \mathbf{R}^{m\times k} \quad \text{and} \quad Y = \begin{bmatrix} y_1 & \cdots & y_n \end{bmatrix} \in \mathbf{R}^{k\times n}$$

be the matrices consisting of the coefficient vectors $x_1, \ldots, x_m \in \mathbf{R}^k$ and $y_1, \ldots, y_n \in \mathbf{R}^k$, respectively. Then the problem (8.33) can be interpreted as finding the maximum likelihood estimate of the model parameters X and Y from the observed data matrix A, given this linear measurement model.

To see this, let $p\colon \mathbf{R} \to \mathbf{R}_+$ be the probability density function of a Gaussian distribution with zero mean and variance σ^2, *i.e.*,

$$p(u) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{u^2}{2\sigma^2}\right)$$

for $u \in \mathbf{R}$. Then the likelihood function $p_{X,Y}\colon \mathbf{R}^{m\times n} \to \mathbf{R}_+$ of the model parameters X and Y given the observed data matrix $A \in \mathbf{R}^{m\times n}$ can be expressed

as

$$\begin{aligned} p_{X,Y}(A) &= \prod_{i=1}^{m}\prod_{j=1}^{n} p(a_{ij} - x_i^T y_j) \\ &= \prod_{i=1}^{m}\prod_{j=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(a_{ij} - x_i^T y_j)^2\right) \\ &= (2\pi\sigma^2)^{-mn/2} \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{m}\sum_{j=1}^{n}(a_{ij} - x_i^T y_j)^2\right). \end{aligned}$$

Therefore, the corresponding log-likelihood objective of this maximum likelihood estimation problem is given by

$$\begin{aligned} l(X,Y) &= \log p_{X,Y}(A) \\ &= -\frac{mn}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{m}\sum_{j=1}^{n}(a_{ij} - x_i^T y_j)^2 \\ &= -\frac{mn}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\|A - XY\|_F^2. \end{aligned}$$

Since the first term in the log-likelihood objective is a constant, maximizing the log-likelihood $l(X,Y)$ is equivalent to minimizing the second term, which is equivalent to solving the problem (8.33).

Feature compression and latent factor estimation

We can interpret the factorized low rank approximation problem (8.33) as *compressing* the n features of the data points $a_1, \ldots, a_m \in \mathbf{R}^n$ into a set of $k < n$ new features. Specifically, the rows $x_1^T, \ldots, x_m^T$ of the matrix $X \in \mathbf{R}^{m\times k}$ can be interpreted as the representations of the data points $a_1, \ldots, a_m$ in the new feature space $\mathbf{R}^k$, and the columns $y_1, \ldots, y_n \in \mathbf{R}^k$ of the matrix $Y \in \mathbf{R}^{k\times n}$ can be interpreted as the coefficients that map the new k features back to the original n features. In this context, the vectors

$$z_i = Y^T x_i \in \mathbf{R}^n, \quad i = 1, \ldots, m,$$

whose transposes form the rows of the approximation matrix $Z = XY \in \mathbf{R}^{m\times n}$ of A, can be considered as the recovered approximations of the data points $a_1, \ldots, a_m \in \mathbf{R}^n$ from the compressed representations $x_1, \ldots, x_m \in \mathbf{R}^k$. A solution of the problem (8.33) therefore corresponds to a feature compression scheme that results in the smallest approximation error $\|A - XY\|_F^2$. This interpretation aligns with our observation in figure 8.9.

Similarly, we can interpret each row x_i^T of the matrix $X \in \mathbf{R}^{m\times k}$ as a vector of k *latent factors* associated with the ith data point $a_i \in \mathbf{R}^n$. Then the problem (8.33) corresponds to finding a group of latent factors $x_1, \ldots, x_m \in \mathbf{R}^k$ that best explain the observed data points $a_1, \ldots, a_m \in \mathbf{R}^n$. If the approximation error $\|A - XY\|_F^2$ is small, then we could say that the latent factors $x_1, \ldots, x_m \in \mathbf{R}^k$ provides a good explanation of the dataset.

Archetype representations

In the factorized low rank approximation problem (8.33), we could think of the *rows* $\tilde{y}_1^T, \ldots, \tilde{y}_k^T$ of the matrix $Y \in \mathbf{R}^{k\times n}$ as k *archetypes* with each capturing the characteristics of one of k extreme samples in the dataset. With this interpretation, each sample $a_i \in \mathbf{R}^n$ in the dataset is then (approximately) represented as a linear combination of these archetypes, with the coefficients given by the corresponding row x_i^T of the matrix $X \in \mathbf{R}^{m\times k}$. The coefficient x_{ij} is sometimes called the *loading* of the ith sample on the jth archetype, which indicates how much the jth archetype contributes to the representation of the ith sample.

In this context, the rows x_i^T of the matrix $X \in \mathbf{R}^{m\times k}$ can be interpreted as the *archetype representations* of the data points $a_1, \ldots, a_m \in \mathbf{R}^n$ in terms of the archetypes $\tilde{y}_1^T, \ldots, \tilde{y}_k^T$. If the archetypes are simple to understand or interpret, then the archetype representations of the data points might provide better intuition about the dataset.

8.3.3 Quadratic regularization

Consider the factorized low rank approximation problem (8.33). Recall that we had the observation in remark 8.3 that the solutions to this problem are not unique. One approach to narrow down the solution set is to add some regularization terms to the problem (8.33), the following *quadratically regularized low rank approximation* (or *quadratically regularized PCA*) problem gives a simple example of this idea:

$$\text{minimize} \quad \|A - XY\|_F^2 + \gamma(\|X\|_F^2 + \|Y\|_F^2), \tag{8.41}$$

where $X \in \mathbf{R}^{m\times k}$ and $Y \in \mathbf{R}^{k\times n}$ are the optimization variables, and $\gamma \geq 0$ is a regularization parameter that controls the strength of the regularization. When $\gamma = 0$, this problem reduces to the original factorized low rank approximation problem (8.33). We could express the problem (8.41) in terms of the rows of X and the columns of Y as

$$\text{minimize} \quad \sum_{i=1}^m \sum_{j=1}^n (a_{ij} - x_i^T y_j)^2 + \gamma \left(\sum_{i=1}^m \|x_i\|_2^2 + \sum_{j=1}^n \|y_j\|_2^2 \right),$$

from which we see that the quadratic regularization term $\gamma(\|X\|_F^2 + \|Y\|_F^2)$ consists in adding Tikhonov regularization to each of the coefficient vectors $x_1, \ldots, x_m \in \mathbf{R}^k$ and $y_1, \ldots, y_n \in \mathbf{R}^k$. Equivalently, we can interpret this quadratic regularization as adding a quadratic penalty $\phi(u) = u^2$ with penalty weight γ on each entry of the variables X and Y, which encourages the matrices X and Y to be (componentwise) small as γ increases.

The problem (8.41) can also be expressed in an equivalent rank constrained form; see exercise 8.4.

Solution of quadratically regularized low rank approximation

The problem (8.41) is a biconvex optimization problem in the variables X and Y, which can therefore be approximately solved via alternate convex search. However,

it is less known that the quadratically regularized low rank approximation problem (8.41) in fact has an analytical solution.

Suppose $\mathbf{rank}\, A = r$, and let $A = U\Sigma V^T$ be a compact SVD of A, where $U \in \mathbf{R}^{m\times r}$ and $V \in \mathbf{R}^{n\times r}$ have orthonormal columns and $\Sigma = \mathbf{diag}(\sigma_1, \ldots, \sigma_r) \in \mathbf{R}^{r\times r}$ is a diagonal matrix with $\sigma_1 \geq \cdots \geq \sigma_r > 0$ being the singular values of A. Then an optimal point $(X^\star, Y^\star)$ of the problem (8.41) is given by

$$X^\star = U_k\tilde{\Sigma}_k^{1/2} \quad \text{and} \quad Y^\star = \tilde{\Sigma}_k^{1/2}V_k^T, \tag{8.42}$$

where $U_k \in \mathbf{R}^{m\times k}$ and $V_k \in \mathbf{R}^{n\times k}$ consist of the first k columns of U and V, respectively, and $\tilde{\Sigma}_k \in \mathbf{R}^{k\times k}$ is a diagonal matrix defined as

$$\tilde{\Sigma}_k = \begin{bmatrix} (\sigma_1-\gamma)_+ & & 0 \\ & \ddots & \\ 0 & & (\sigma_k-\gamma)_+ \end{bmatrix} \in \mathbf{R}^{k\times k},$$

with $(\sigma_i - \gamma)_+ = \max\{0, \sigma_i - \gamma\}$ for $i = 1, \ldots, k$. Obviously, when $\gamma = 0$, the optimal point given by (8.42) reduces to the solution of the original factorized low rank approximation problem (8.33), which is given by (8.35).

We also have the observation that if $X^\star$ and $Y^\star$ are optimal points of the problem (8.41), then for any *orthogonal* matrix $W \in \mathbf{R}^{k\times k}$, we have

$$\begin{aligned} & \|A - (X^\star W)(W^T Y^\star)\|_F^2 + \gamma\left(\|X^\star W\|_F^2 + \|W^T Y^\star\|_F^2\right) \\ = \;& \|A - X^\star Y^\star\|_F^2 + \gamma(\|X^\star\|_F^2 + \|Y^\star\|_F^2), \end{aligned}$$

where the equality in the primary matrix factorization objective is due to the orthogonality of W, *i.e.*, $WW^T = I$, and the equalities in the regularization terms follow from the fact that the Frobenius norm is invariant under orthogonal transformations (see A.3.1, page 351). This implies that, on the one hand, the quadratically regularized problem (8.41) still has nonunique solutions. On the other hand, the solution set is indeed much smaller than that of the original problem (8.33).

Moreover, compared to the original problem (8.33), the set of optimal points of the problem (8.41) is bounded. To see this, we first notice that the objective value of the problem (8.41) at the feasible point $(X, Y) = (0, 0)$ is given by $\|A\|_F^2$. Since the objective value at any optimal point $(X^\star, Y^\star)$ must be no larger than $\|A\|_F^2$, we have

$$\|A - X^\star Y^\star\|_F^2 + \gamma(\|X^\star\|_F^2 + \|Y^\star\|_F^2) \leq \|A\|_F^2,$$

which implies that

$$\|X^\star\|_F^2 + \|Y^\star\|_F^2 \leq \frac{1}{\gamma}\|A\|_F^2.$$

This means that both matrices $X^\star$ and $Y^\star$ at any optimal point of the problem (8.41) must be bounded. In particular, optimal points of the problem (8.41) must lie in the Euclidean ball

$$\left\{(X, Y) \in \mathbf{R}^{m\times k} \times \mathbf{R}^{k\times n} \;\middle|\; \|X\|_F^2 + \|Y\|_F^2 \leq \frac{1}{\gamma}\|A\|_F^2\right\}$$

centered at the origin in $\mathbf{R}^{m\times k} \times \mathbf{R}^{k\times n}$ with radius $\sqrt{(1/\gamma)\|A\|_F^2}$.

Geometric interpretation

According to the solution expression given by (8.42) of the quadratically regularized low rank approximation problem (8.41), the best rank-k approximation matrix $Z^\star = X^\star Y^\star$ of the original data matrix A is given by

$$Z^\star = U_k \tilde{\Sigma}_k V_k^T = U_k \Sigma_k D(\gamma) V_k^T, \tag{8.43}$$

where $\Sigma_k = \mathbf{diag}(\sigma_1, \ldots, \sigma_k) \in \mathbf{R}^{k\times k}$ is the diagonal matrix consisting of the top k singular values of A, and the *shrinkage matrix* $D(\gamma) \in \mathbf{R}^{k\times k}$ is defined as

$$D(\gamma) = \begin{bmatrix} (\sigma_1 - \gamma)_+/\sigma_1 & & 0 \\ & \ddots & \\ 0 & & (\sigma_k - \gamma)_+/\sigma_k \end{bmatrix},$$

which is parameterized by the regularization weight $\gamma \geq 0$ and has diagonal entries in the interval $[0, 1]$. Noticing that

$$AV_k = U\Sigma V^T V_k = U_k \Sigma_k,$$

we have

$$Z^\star = AV_k D(\gamma) V_k^T,$$

and therefore, each row of the approximation matrix $Z^\star$ can be expressed as

$$z_i^{\star T} = a_i^T V_k D(\gamma) V_k^T, \quad i = 1, \ldots, m,$$

where a_i^T is the ith row of the data matrix A. This implies that the solution $Z^\star$ of the quadratically regularized problem (8.41) is obtained by projecting the data points $a_1, \ldots, a_m \in \mathbf{R}^n$ onto the subspace spanned by the top k right singular vectors $v_1, \ldots, v_k \in \mathbf{R}^n$ of A, and then shrinking the resulting projection coefficients along each singular-vector direction according to the diagonal entries of $D(\gamma)$. When $\gamma = 0$, the shrinkage matrix $D(\gamma) = I$ is the identity matrix, so there is no shrinkage on the projection coefficients, and thus the best rank-k approximation $Z^\star$ from (8.41) reduces to the solution from the original problem (8.33). When $\gamma > 0$, the shrinkage factors in $D(\gamma)$ are all less than 1, so the projected data points $z_1, \ldots, z_m \in \mathbf{R}^n$ are shrunk towards zero. In particular, when γ is sufficiently large such that $\gamma \geq \sigma_1$, all shrinkage factors in $D(\gamma)$ become zero, and thus all data points are projected to the origin.

Figure 8.11 illustrates this idea on an example dataset in $\mathbf{R}^2$, where each plot corresponds to a solution of the quadratically regularized low rank approximation problem (8.41) with $k = 1$, under a different value of the regularization parameter $\gamma \geq 0$. When $\gamma = 0$, the best quadratically regularized rank-1 approximation (shown circle) of the original data points (shown crosses) is the same as the solution of the original low rank approximation problem (8.33). As γ increases, all projected data points are shrunk towards zero. When γ is sufficiently large (as shown in the bottom right plot), all projected data points are shrunk to the origin. Note that the direction of the one-dimensional subspace corresponding to the best rank-1 approximation (indicated by the dashed line) does not change with γ, which is consistent with our observation from the solution expression (8.43).

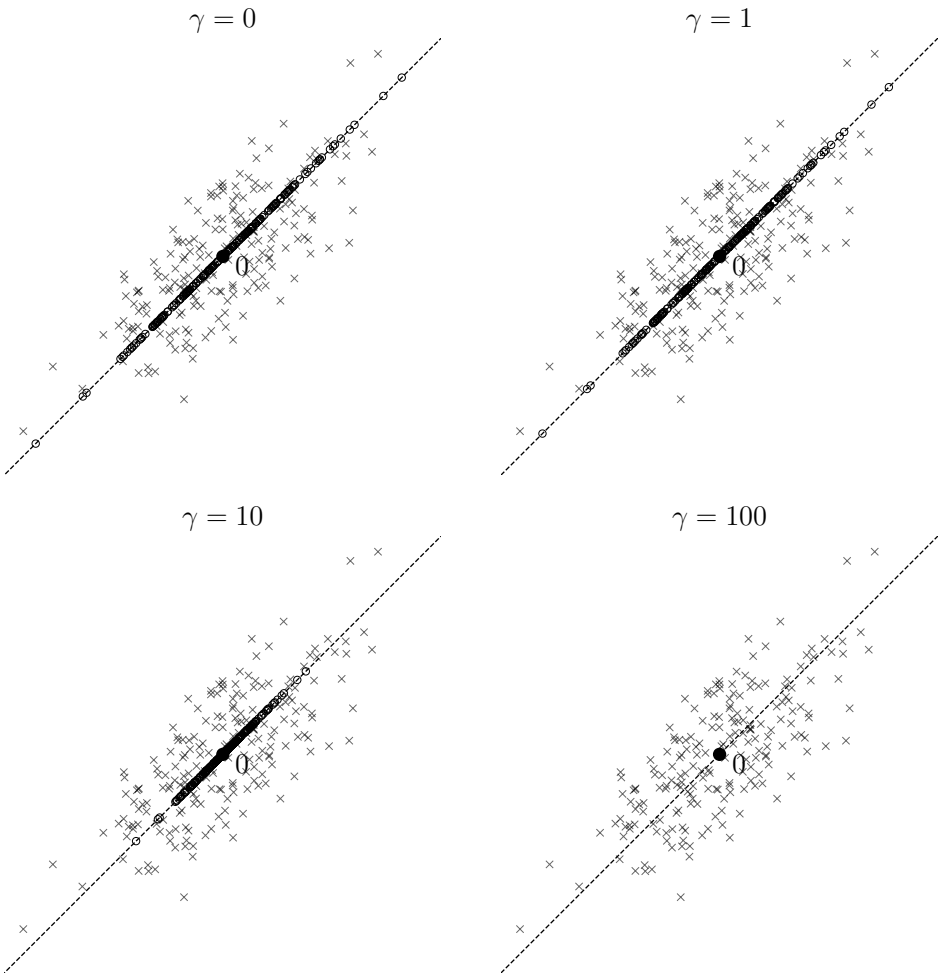

Figure 8.11 Each plot corresponds to a solution of the quadratically regularized low rank approximation problem (8.41) on a dataset $A \in \mathbf{R}^{200\times 2}$ with $k = 1$, under a different value of the regularization parameter $\gamma \geq 0$. The original data points are shown as crosses, and the projected data points are shown as circles. The dashed line in each plot indicates the one-dimensional subspace corresponding to the best rank-1 approximation, onto which the data points are projected.

Probabilistic interpretation

The quadratically regularized low rank approximation problem (8.41) can be interpreted as a maximum a posteriori estimation problem in a probabilistic model that extends the linear measurement model for the original problem (8.33).

Specifically, we assume that each entry $a_{ij} \in \mathbf{R}$ of the data matrix $A \in \mathbf{R}^{m\times n}$ is generated from the linear measurement model

$$a_{ij} = x_i^T y_j + \epsilon_{ij}, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n,$$

where $x_i \in \mathbf{R}^k$ and $y_j \in \mathbf{R}^k$ are the coefficient vectors associated with the ith row and the jth column of A, respectively, *i.e.*,

$$X = \begin{bmatrix} x_1^T \\ \vdots \\ x_m^T \end{bmatrix} \in \mathbf{R}^{m\times k} \quad \text{and} \quad Y = \begin{bmatrix} y_1 & \cdots & y_n \end{bmatrix} \in \mathbf{R}^{k\times n}.$$

The noise $\epsilon_{ij} \in \mathbf{R}$ are IID standard Gaussian random variables. In addition, we assume that the entries of the matrix variables X and Y are IID Gaussian random variables with mean zero and variance σ^2, *i.e.*,

$$x_{ij} \sim \mathcal{N}(0, \sigma^2), \quad i = 1, \ldots, m, \quad j = 1, \ldots, k,$$

and

$$y_{ij} \sim \mathcal{N}(0, \sigma^2), \quad i = 1, \ldots, k, \quad j = 1, \ldots, n.$$

Recall (from §5.4) that the problem of maximizing the posterior distribution of the model parameters X and Y given the observed data matrix A can be expressed as

$$\text{maximize} \quad \log p_{A|X,Y}(X, Y, A) + \log p_{X,Y}(X, Y) \tag{8.44}$$

with variables $X \in \mathbf{R}^{m\times k}$ and $Y \in \mathbf{R}^{k\times n}$, where $p_{A|X,Y}$ is the likelihood function of the model parameters X and Y given the observed data A, and $p_{X,Y}$ is the prior of the model parameters. We have already seen on page 322 that the first log-likelihood term in (8.44) of this linear measurement model with IID standard Gaussian noise is given by

$$\log p_{A|X,Y}(X, Y, A) = -\frac{mn}{2}\log(2\pi) - \frac{1}{2}\|A - XY\|_F^2.$$

Let $p\colon \mathbf{R} \to \mathbf{R}_+$ be the probability density function of the Gaussian distribution $\mathcal{N}(0, \sigma^2)$, the second log-prior term in (8.44) can be expressed as

$$\begin{aligned}
& \log p_{X,Y}(X, Y) \\
= \; & \sum_{i=1}^{m}\sum_{j=1}^{k} \log p(x_{ij}) + \sum_{i=1}^{k}\sum_{j=1}^{n} \log p(y_{ij}) \\
= \; & -\frac{mk}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{m}\sum_{j=1}^{k} x_{ij}^2 - \frac{kn}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{k}\sum_{j=1}^{n} y_{ij}^2 \\
= \; & -\frac{mk+kn}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}(\|X\|_F^2 + \|Y\|_F^2).
\end{aligned}$$

Therefore, the problem (8.44) of maximizing the posterior distribution of the model parameters X and Y given the observed data matrix A is equivalent to

$$\text{minimize} \quad \|A - XY\|_F^2 + (1/\sigma^2)(\|X\|_F^2 + \|Y\|_F^2),$$

which is exactly the quadratically regularized low rank approximation problem (8.41) with regularization parameter $\gamma = 1/\sigma^2$.

When the variance σ^2 of the Gaussian prior is small, we assign higher probabilities to smaller values of the entries of X and Y, which corresponds to stronger regularization in the problem (8.41). On the other hand, as $\sigma^2 \to \infty$, the Gaussian prior on the model parameters becomes the noninformative prior that assigns equal probabilities to all possible values of X and Y, which corresponds to the case when there is no regularization in the problem (8.41).

8.3.4 Matrix completion

A generalization of the low rank approximation problem (8.32) is to the case when only a subset of the entries in the data matrix $A \in \mathbf{R}^{m\times n}$ are observed, and we want to find a low rank approximation of A that best fits the observations. Let $\mathcal{I} \subseteq \{1,\dots,m\}\times\{1,\dots,n\}$ be the index set where $a_{ij} \in \mathbf{R}$ with $(i,j) \in \mathcal{I}$ denote all observed entries of A, the problem of *low rank matrix completion* can be formulated as

$$\begin{array}{ll} \text{minimize} & \sum_{(i,j)\in\mathcal{I}} (a_{ij} - z_{ij})^2 \\ \text{subject to} & \mathbf{rank}\, Z \leq k, \end{array} \tag{8.45}$$

where $Z \in \mathbf{R}^{m\times n}$ is the optimization variable, z_{ij} is the (i,j)th entry of Z, and $k < n$ is the given maximum rank. The problem (8.45) consists in finding a rank-k matrix Z that best fits the observed entries of A in the least squares sense.

Similarly, we could write the problem of low rank matrix completion in its factorized form as

$$\text{minimize} \quad \sum_{(i,j)\in\mathcal{I}} (a_{ij} - x_i^T y_j)^2 \tag{8.46}$$

with variables $X \in \mathbf{R}^{m\times k}$ and $Y \in \mathbf{R}^{k\times n}$, where x_i^T is the ith row of X and y_j is the jth column of Y. This formulation results in a unconstrained biconvex optimization problem so that we don't need to deal with the hard rank constraint in (8.45). Moreover, quadratic regularization terms as in (8.41) can also be incorporated into the problem (8.46) to narrow down the size of the solution set.

Interpretations

In the context of data analysis, the problem (8.45) (and (8.46)) can be think of as performing PCA on an incomplete dataset. Depending on the size of the index set $\mathcal{I}$ of the observed entries in the data matrix A, this can be interpreted in different ways.

When the index set $\mathcal{I}$ is large, *i.e.*, only a few entries in the data matrix A are missing, the typical approach in practice for handling the missing data is to just remove the data points, *i.e.*, the rows of A, that contain missing entries, and use the remaining data points for subsequent analysis. If instead we decide *not* to remove

any data point but solve the problem (8.45), then we can interpret the solution of (8.45) as a way to *impute* the missing entries of A (which we would normally discard) based on the observed entries.

On the other hand, when $\mathcal{I}$ is small, *i.e.*, most entries in the data matrix A are missing, it is not possible to simply remove the data points with missing entries, since we might discard the entire dataset. In this case, we have to recover the missing entries of A according to the observations in $\mathcal{I}$ by solving the problem (8.45), presuming that the data matrix A has a low rank structure.

Solution methods

In general, there is no analytical solution for the matrix completion problem (8.45) (or (8.46)), and we have to resort to some kind of heuristic to find an approximate solution. One simple heuristic is to recognize the biconvex structure of the factorized formulation (8.46) and apply alternate convex search, so that we could obtain a partially optimal point of the problem (8.46).

Another class of heuristics approximates the nonconvex rank constrained problem (8.45) with a convex regularized version (which can therefore be solved efficiently), given by

$$\text{minimize} \quad \sum_{(i,j)\in\mathcal{I}} (a_{ij} - z_{ij})^2 + \gamma \|Z\|_*, \tag{8.47}$$

where $Z \in \mathbf{R}^{m\times n}$ is the optimization variable and $\gamma > 0$ is the regularization coefficient. The norm $\|\cdot\|_*$ is the *nuclear norm* of a matrix, defined as

$$\|Z\|_* = \sigma_1 + \cdots + \sigma_n,$$

where $\sigma_1 \geq \cdots \geq \sigma_n \geq 0$ are the (possibly zero) singular values of Z (assuming that $m \geq n$). Recall (from §5.5.3) that the nuclear norm regularizer is an efficient convex surrogate for the rank regularization function. Hence, as the regularization parameter γ increases, solution of the problem (8.47) is more likely to have a smaller nuclear norm, and thus a smaller rank. In particular, when γ is sufficiently large, the solution of this problem would be the zero matrix, which has rank zero. A numerical example of this idea is presented in example 5.9, page 188.

8.4 Generalized low rank models

Now we present some ideas of customizing the low rank approximation problem (8.32) (or (8.33)) for different application scenarios and incorporating different types of prior knowledge.

8.4.1 Robust low rank approximation

Recall that the ordinary low rank approximation problem (8.32) corresponds to applying a quadratic penalty $\phi(u) = u^2$ on the approximation error $a_{ij} - z_{ij}$ between each entry a_{ij} of the data matrix $A \in \mathbf{R}^{m\times n}$ and the corresponding entry z_{ij} of the approximation $Z \in \mathbf{R}^{m\times n}$ (where a_i^T and z_i^T are the ith rows of A and Z,

respectively). In the most general case, we can replace this quadratic penalty with any other penalty function $\phi\colon \mathbf{R} \to \mathbf{R}$, such as those presented in §4.1. Here we discuss one particular choice of penalty that leads to a robust version of the low rank approximation problem.

Low rank approximation with absolute value penalty

We have seen in §4.1.3 that the quadratic penalty $\phi(u) = u^2$ is sensitive to large approximation errors due to, *e.g.*, outliers in the data. A simple idea of increasing the robustness of the low rank approximation is to replace the quadratic penalty with the absolute value penalty $\phi(u) = |u|$, which leads to the problem

$$\begin{array}{ll} \text{minimize} & \|A - Z\|_{\text{sav}} = \sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij} - z_{ij}| \\ \text{subject to} & \mathbf{rank}\, Z \leq k \end{array} \tag{8.48}$$

with variable $Z \in \mathbf{R}^{m \times n}$, where $\|\cdot\|_{\text{sav}}$ is the *sum-absolute-value norm* of matrices (see example A.2, page 353). Since the absolute value penalty is less sensitive to large approximation errors than the quadratic penalty, low rank approximation from the problem (8.48) is expected to be less sensitive to outliers in the data than that from the original problem (8.32).

Unlike the problem (8.32), there is no analytical solution for the problem (8.48), so we usually approximately solve it via alternate minimization on the factorized biconvex formulation of this problem, given by

$$\text{minimize} \quad \|A - XY\|_{\text{sav}} = \sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij} - x_i^T y_j|, \tag{8.49}$$

where $X \in \mathbf{R}^{m \times k}$ and $Y \in \mathbf{R}^{k \times n}$ are the optimization variables, and x_i^T is the ith row of X and y_j is the jth column of Y.

Example 8.6 *Low rank approximation with outliers.* We compare the nominal low rank approximation problem (8.32) and the robust problem (8.48) for $k = 1$ on a dataset in $\mathbf{R}^2$ that contains some outliers. Figure 8.12 plots the original data points as crosses, where the outliers are shown at the top left and bottom right of the figure.

The best approximated data points from the nominal problem (8.32) are shown as circles in figure 8.12. It is seen that the corresponding subspace (shown as a dashed line) in which the approximated data points lie is significantly affected by the outliers. In particular, this subspace is almost orthogonal to the dominant direction of variation among the majority of the data points. This occurs because the direction that captures the largest variance of the whole dataset is strongly distorted toward the line connecting the outliers. On the other hand, the approximations from the robust problem (8.48) (shown as triangles) is much less affected by the outliers, and the corresponding subspace in which the approximated data points lie is more aligned with the dominant direction of variation among the majority of the data points.

Figure 8.13 plots the histogram of Euclidean distances between the original data points a_i and their approximations z_i from the nominal problem (8.32) and its robust version (8.48) for $i = 1, \ldots, m$, respectively. Generally speaking, the distribution of the distances corresponding to the robust problem (8.48) (shown darker) is more concentrated towards zero than that of the nominal problem (8.32). However, there are

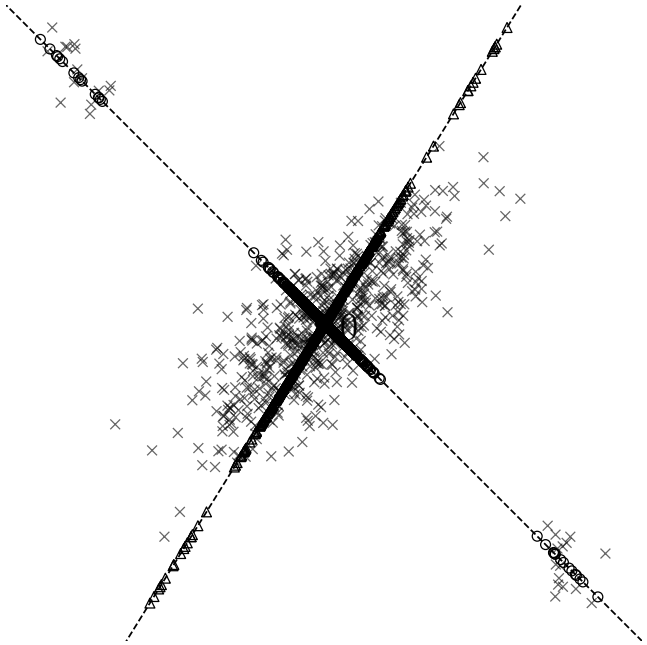

Figure 8.12 The cross marks indicate the original data points in $\mathbf{R}^2$. Some of the points are outliers, which are shown at the top left and bottom right of the figure. The circles and triangles indicate the low dimensional approximated data points obtained from the nominal low rank approximation problem (8.32) and its robust version (8.48) for $k = 1$, respectively. The corresponding one-dimensional subspaces in which the approximations lie are shown as dashed lines.

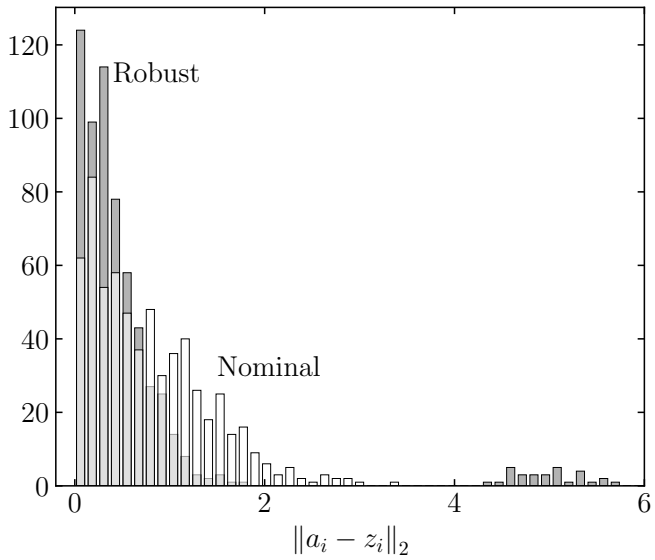

Figure 8.13 Histogram of Euclidean distances between the original data points a_i in figure 8.12 and their approximations z_i from the nominal problem (8.32) (shown darker) and its robust version (8.48) for $i = 1, \ldots, m$, respectively.

several very large distance values (between 4 and 6) for the robust approximation from (8.48) that correspond to the outliers in the data, while the distance amplitudes corresponding to the nominal approximation from (8.32) are all relatively small (roughly below 3.5).

Robust PCA

In practice, the commonly used *robust low rank approximation*, or *robust principal component analysis*, is given by the following quadratically regularized version of the absolute value penalty low rank approximation problem (8.49):

$$\text{minimize} \quad \|A - XY\|_{\text{sav}} + \gamma(\|X\|_F^2 + \|Y\|_F^2) \tag{8.50}$$

where $X \in \mathbf{R}^{m\times k}$ and $Y \in \mathbf{R}^{k\times n}$ are the optimization variables, and $\gamma \geq 0$ is the regularization parameter. Using the similar arguments as we have seen exercise 8.4, we could show that the problem (8.50) is equivalent to the problem (8.48) with an additional nuclear norm regularization, *i.e.*,

$$\begin{array}{ll} \text{minimize} & \|A - Z\|_{\text{sav}} + 2\gamma\|Z\|_* \\ \text{subject to} & \mathbf{rank}\, Z \leq k, \end{array} \tag{8.51}$$

where $Z \in \mathbf{R}^{m\times n}$ is the variable and $\|\cdot\|_*$ is the nuclear norm.

It happens sometimes in practice that people remove the rank constraint in (8.51) and introduce the auxiliary variable $S = A - Z$ representing the residual matrix, so that we have the following simpler formulation of the robust PCA:

$$\begin{array}{ll} \text{minimize} & \|S\|_{\text{sav}} + 2\gamma\|Z\|_* \\ \text{subject to} & S + Z = A \end{array} \tag{8.52}$$

with variables $S, Z \in \mathbf{R}^{m\times n}$. This formulation can be interpreted as decomposing the original data matrix A into a low rank component Z and a sparse component S, where the low rank component captures the dominant variation in the data, and the sparse component captures the outliers. One very important advantage of the robust PCA formulation (8.52) is that, unlike the problems (8.50) and (8.51), it is a convex optimization problem so can be solved efficiently without any heuristic. However in this case, we have to control the rank of the low rank component Z by tuning the regularization parameter γ instead of directly specifying the maximum rank k as in the problems (8.50) and (8.51).

See also exercise 8.5 for another robust version of the low rank approximation problem that uses the Huber penalty to combine both characteristics from the problems (8.32) and (8.48).

8.4.2 Structural regularization and constraints

Consider the factorized formulation of the low rank approximation problem given by (8.33). This section discusses how different types of structural properties can be induced in the approximation solutions by incorporating suitable regularization terms and constraints into the problem (8.33). These ideas are readily adapted to low rank approximation problems with different cost functions for the approximation error.

Note that, unlike the discussions above, here we *do not* assume the data matrix $A \in \mathbf{R}^{m\times n}$ is always standardized. We will see later that for many generalized low rank models, it is actually often desirable to work with the original data matrix A without standardization, so that the approximations can be more interpretable in the original feature space.

Nonnegative matrix factorization

Consider the problem of low rank approximation with nonnegativity constraints on the factor matrices X and Y, given by

$$\begin{array}{lll} \text{minimize} & \|A - XY\|_F^2 = \sum_{i=1}^m \sum_{j=1}^n (a_{ij} - x_i^T y_j)^2 & \\ \text{subject to} & x_i \succeq 0, \quad i = 1, \ldots, m & \\ & y_i \succeq 0, \quad i = 1, \ldots, n, & \end{array} \tag{8.53}$$

where $X \in \mathbf{R}^{m\times k}$ and $Y \in \mathbf{R}^{k\times n}$ are the optimization variables, $x_1^T, \ldots, x_m^T$ are the rows of X, and $y_1, \ldots, y_n$ are the columns of Y. The problem (8.53) is known as the *nonnegative matrix factorization* (NMF) problem. In the most general case, the biconvex NMF problem (8.53) does not have an analytical solution, so it is often approximately solved via the alternate minimization heuristic.

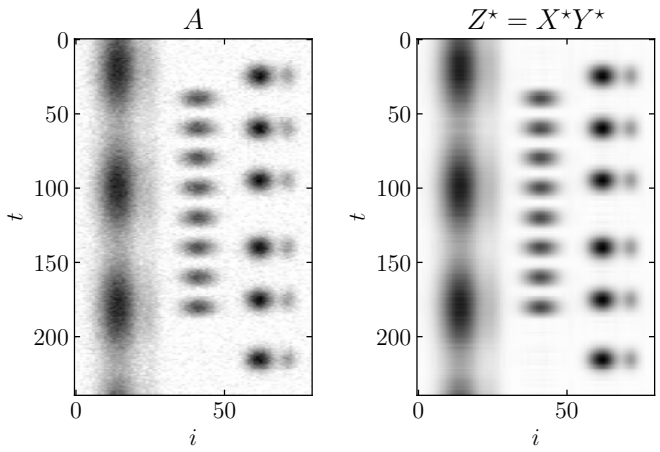

Figure 8.14 *Left.* A nonnegative data matrix $A \in \mathbf{R}_+^{m\times n}$ with $m = 240$ and $n = 80$, where the darker color indicates larger values of the entries in A. *Right.* The approximated data matrix $Z^\star = X^\star Y^\star$ with rank $k = 3$ obtained from the NMF problem (8.54).

The NMF problem (8.53) often appears in applications where the data matrix A contains nonnegative entries, *e.g.*, images, texts, audio signals, and gene expression data, and we want to find a low rank approximation of A that is also nonnegative and thus more interpretable. From an archetype representation perspective, each data point $a_i \in \mathbf{R}^n$ (whose transpose is the ith row of A) is approximated by a nonnegative linear (*i.e.*, conic) combination of the rows of Y (which can be interpreted as k archetypes in the original feature space $\mathbf{R}^n$), where the nonnegative coefficients are given by the corresponding row of X. Geometrically, this corresponds to finding a convex cone spanned by the rows of Y (which lies in a k-dimensional subspace of $\mathbf{R}^n$), so that the sum of Euclidean distances between the data points and their projections onto this convex cone are minimized.

Example 8.7 *NMF in signal processing.* Consider a nonnegative data matrix $A \in \mathbf{R}_+^{m\times n}$, where the tth row a_t^T of A corresponds to a signal vector in $\mathbf{R}^n$ observed at a particular time step for $t = 1, \ldots, m$, and the ith column of A represents a nonnegative time series of the ith feature at all time steps, such as the energy in a frequency band, for $i = 1, \ldots, n$. Figure 8.14 shows an example of such a data matrix A with $m = 240$ and $n = 80$ on the left. Suppose we want to find a nonnegative approximation of A with rank $k = 3$ that captures the dominant patterns in the signals. To achieve this, consider the NMF problem given by

$$\begin{array}{ll} \text{minimize} & \|A - XY\|_F^2 = \sum_{t=1}^m \sum_{i=1}^n \left(a_{ti} - x_t^T y_i\right)^2 \\ \text{subject to} & x_t \succeq 0, \quad t = 1, \ldots, m \\ & y_i \succeq 0, \quad i = 1, \ldots, n, \end{array} \tag{8.54}$$

where $X \in \mathbf{R}^{m\times k}$ and $Y \in \mathbf{R}^{k\times n}$ are the optimization variables with $k = 3$, x_t^T is the tth row of X, and y_i is the ith column of Y.

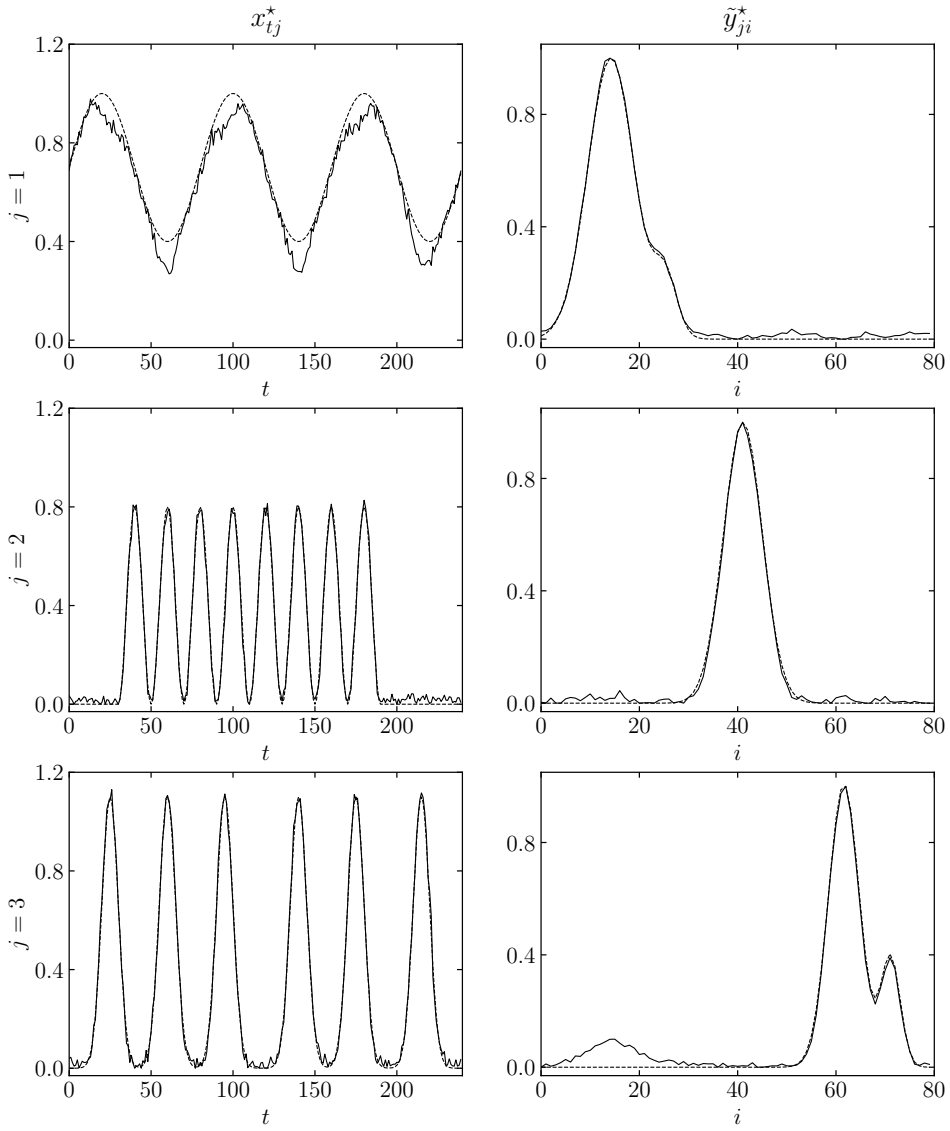

Figure 8.15 Plots of the factor matrices $X^\star \in \mathbf{R}^{m\times k}$ and $Y^\star \in \mathbf{R}^{k\times n}$ obtained from the NMF problem (8.54) with data matrix A shown in figure 8.14. *Left.* The jth column $(x^\star_{1j}, \ldots, x^\star_{mj}) \in \mathbf{R}^m$ of $X^\star$ plotted as a function of time $t = 1, \ldots, m$ for $j = 1, \ldots, k$. *Right.* The jth row $\tilde{y}_j^{\star T}$ of $Y^\star$ plotted as a function of the feature index $i = 1, \ldots, n$ for $j = 1, \ldots, k$. The ground truth used to generate the original data matrix A is also shown as dashed lines for reference.

Suppose $(X^\star, Y^\star)$ is a (partially; see §3.1.3) optimal point of the NMF problem (8.54) obtained via alternate convex search. The corresponding approximated data matrix $Z^\star = X^\star Y^\star$ of A is shown on the right of figure 8.14. Let $\tilde{y}_j^{\star T}$ denote the jth row of $Y^\star$ for $j = 1, \ldots, k$, then $\tilde{y}_1^\star, \ldots, \tilde{y}_k^\star \in \mathbf{R}^n$ represent the k archetypes in the original feature space obtained from (8.54). For fixed $j = 1, \ldots, k$, the vector $(x_{1j}^\star, \ldots, x_{mj}^\star) \in \mathbf{R}^m$ (*i.e.*, the jth column of $X^\star$) represents the time series of the nonnegative coefficients for the jth archetype across all time steps, or in other words, the loading of the jth archetype in the approximated data for all $t = 1, \ldots, m$. These results are shown in figure 8.15, where the jth column $(x_{1j}^\star, \ldots, x_{mj}^\star)$ of $X^\star$ is plotted as a function of time $t = 1, \ldots, m$ for $j = 1, \ldots, k$ on the left, and the jth row $\tilde{y}_j^{\star T}$ of $Y^\star$ is plotted as a function of the feature index $i = 1, \ldots, n$ on the right.

Sparse PCA

Based on the archetype interpretation of the low rank approximation problem (8.33) with data matrix $A \in \mathbf{R}^{m \times n}$, the rows $\tilde{y}_1^T, \ldots, \tilde{y}_k^T$ of the factor matrix $Y \in \mathbf{R}^{k \times n}$ represent the k archetypes in the original feature space $\mathbf{R}^n$, and the rows $x_1^T, \ldots, x_m^T$ of $X \in \mathbf{R}^{m \times k}$ are then the archetype representation coefficients for the data points $a_1^T, \ldots, a_m^T$ that form the rows of A. Then it is natural to ask if we can find a low rank approximation of A such that only a few of k archetypes are involved in the approximation of each data point. This can be achieved by incorporating a sparsity-inducing regularization term on the factor matrix X into the problem (8.33), *i.e.*, we can consider the problem

$$\text{minimize} \quad \|A - XY\|_F^2 + \gamma \textstyle\sum_{i=1}^m \|x_i\|_1 \tag{8.55}$$

with variables $X \in \mathbf{R}^{m \times k}$ and $Y \in \mathbf{R}^{k \times n}$, where $x_1^T, \ldots, x_m^T$ are the rows of X, and $\gamma \geq 0$ is the regularization parameter. The problem (8.55) is sometimes referred to as the *sparse principal component analysis* problem. The ℓ_1-norm regularization function in (8.55) is a convex surrogate for the cardinality function, which hence encourages the row vectors of X (and thus X itself) to be entrywise sparse, so that only a few archetypes are involved in the approximation of each data point. As the regularization parameter γ increases, the approximations of the data points from the problem (8.55) are more likely to involve fewer archetypes, and thus possibly more interpretable.

It is possible to combine different types of regularization terms and constraints in the same low rank approximation problem of the form (8.33) to obtain a mixture of different structural properties in the factor matrices X and Y. For example, consider the problem

$$\begin{array}{ll} \text{minimize} & \|A - XY\|_F^2 + \gamma \|X\|_{\text{sav}} \\ \text{subject to} & \|Y\|_F^2 \leq \beta \end{array}$$

with variables $X \in \mathbf{R}^{m \times k}$ and $Y \in \mathbf{R}^{k \times n}$. This problem is sometimes called the *dictionary learning* problem, where the rows of the factor matrix Y are interpreted as a dense dictionary of k norm-bounded archetypes, or atoms, in the original feature space $\mathbf{R}^n$, and the factor matrix X is interpreted as the corresponding sparse codes for the data points in terms of the dictionary. The total size of the dictionary used

in many applications is often very large ($k \gg n$), but each sample is represented using only a small number of the atoms.

Feature selection

Similar ideas as in sparse PCA apply to the factor matrix $Y \in \mathbf{R}^{k\times n}$ to encourage the archetypes, *i.e.*, the rows $\tilde{y}_1^T, \ldots, \tilde{y}_k^T$ of Y, to be sparse. Specifically, we want to find a group of k archetypes in the original feature space $\mathbf{R}^n$, such that only a few of the original n features are involved. In other words, we want to find a factor matrix Y that is columnwise sparse so that only a subset of the original features are selected in the approximation. This leads to the following problem formulation:

$$\text{minimize} \quad \|A - XY\|_F^2 + \gamma \textstyle\sum_{i=1}^n \|y_i\|_2 \tag{8.56}$$

with variables $X \in \mathbf{R}^{m\times k}$ and $Y \in \mathbf{R}^{k\times n}$, where $y_1, \ldots, y_n \in \mathbf{R}^k$ are the *columns* of Y, and $\gamma \geq 0$ is the regularization parameter. It is important to note that here we use the ℓ_2-norm regularization function without the square. The regularization term $\sum_{i=1}^n \|y_i\|_2$ in the problem (8.56) can be interpreted as the ℓ_1-norm of the vector of ℓ_2-norms of the columns of Y, given by $(\|y_1\|_2, \ldots, \|y_n\|_2) \in \mathbf{R}^n$, which therefore induces sparsity in the latter vector and thus encourages the matrix of Y to be columnwise sparse. (See also §5.5.2, page 184.)

Bibliographical notes

Traditional presentation of mixture models and latent factor estimation from the perspective of probabilistic modeling and inference can be found in many textbooks on machine learning, such as Bishop [Bis06, chapters 9, 12, and 14], Murphy [Mur22, chapter 20], and Murphy [Mur23, chapter 28]. See also Zhu *et al.* [ZYHB25] for more examples of mixture models.

The original idea of clustering and the k-means algorithm (algorithm 8.1) dates back to Steinhaus in 1956 [Ste56]. It was also independently proposed by Lloyd in 1957 and is hence sometimes called the *Lloyd's algorithm*, although his work was not published until the 1980s [Llo82]. The name *k-means* was first used by MacQueen [Mac67] since the 1960s. NP-hardness of the clustering problem (8.4) was shown by Aloise *et al.* [ADHP09].

A proof of the SVD solution (8.34) to the low rank approximation problem (8.32) was given by Eckart and Young in 1936 [EY36]. See also Mirsky [Mir60] for a more general result on the low rank approximation of matrices under unitarily invariant norms, which includes the Frobenius norm as a special case. A proof of the analytical solution (8.42) to the quadratically regularized low rank approximation problem (8.41) can be found in Mazumder *et al.* [MHT10] and Udell *et al.* [UHZB16, §A.1.1]. Some references about matrix completion and low rank matrix recovery are given in the bibliographical notes on page 190.

The idea of robust PCA in the form of the problem (8.52) was presented in the work of Candès *et al.* [CLMW11], Wright *et al.* [WGR+09], and Xu *et al.* [XCS10, XCS12]. NMF has a long history in chemometrics, under the name *self modeling curve resolution* [LS71]. Some early applications of NMF in signal processing and time series analysis include [PTAK91], [PT94], and [APTJ95]. The name *nonnegative matrix factorization* became widely known after Lee and Seung [LS99, LS00]. See the monograph by Udell *et al.* [UHZB16] for a complete review of generalized low rank models and matrix completion; see also the references on page 15.

Exercises

8.1 *Optimality of the probability simplex relaxation.* The problem (8.6) can be abstractly written as the linear program

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^m c_i^T z_i \\ \text{subject to} & z_i \in \mathbf{conv}\{e_1, \ldots, e_k\}, \quad i = 1, \ldots, m, \end{array} \tag{8.57}$$

where $z_i \in \mathbf{R}^k$ are the variables and $c_i \in \mathbf{R}^k$ are fixed problem data for $i = 1, \ldots, m$. Show that if the vectors $c_1, \ldots, c_m \in \mathbf{R}^k$ have unique minimum components, then the problem (8.57) is equivalent to the problem

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^m c_i^T z_i \\ \text{subject to} & z_i \in \{e_1, \ldots, e_k\}, \quad i = 1, \ldots, m, \end{array}$$

where $e_1, \ldots, e_k \in \mathbf{R}^k$ are the standard basis vectors. In other words, if the vectors $c_1, \ldots, c_m \in \mathbf{R}^k$ have unique minimum components, then an optimal point of the probability simplex constrained problem (8.57) must satisfy $z_i \in \{e_1, \ldots, e_k\}$ for $i = 1, \ldots, m$.

8.2 [ZYHB25] *Linear output hidden Markov models.* Suppose we have a dataset $(x(t), y(t))$, $t = 1, \ldots, m$, where $x(t) \in \mathbf{R}^n$ is the feature vector at time t and $y(t) \in \mathbf{R}$ is the corresponding response, observed from a k-state hidden Markov model with linear outputs. In particular, it is assumed that the output $y(t)$ at each time step is generated from a linear measurement model, given by

$$y(t) = x(t)^T \theta_{z(t)} + \epsilon(t), \quad t = 1, \ldots, m,$$

where $z(t) \in \{1, \ldots, k\}$ is the hidden state at time t, $\theta_{z(t)} \in \mathbf{R}^n$ is the coefficient vector associated with the state $z(t)$, and $\epsilon(t) \in \mathbf{R}$ is standard Gaussian noise. The hidden states $z(1), \ldots, z(m)$ are generated by a Markov process with transition matrix $P \in \mathbf{R}^{k \times k}$, *i.e.*,

$$P_{ij} = \mathbf{prob}(z(t) = j \mid z(t-1) = i), \quad i, j = 1, \ldots, k, \quad t = 2, \ldots, m.$$

We want to estimate the model parameters $\theta_1, \ldots, \theta_k \in \mathbf{R}^n$ and $P \in \mathbf{R}^{k \times k}$ from the dataset $(x(t), y(t))$, $t = 1, \ldots, m$. Describe a procedure to estimate these parameters via optimization. The estimation does not have to be exact, but should be based on solving some convex optimization problems. You need to clearly specify the optimization problems to be solved, and how the solutions of these problems are used to estimate the model parameters.

8.3 *Variance maximization interpretation of low rank approximation.* In this exercise, we complete the proof in remark 8.4 for the case when $k > 1$. We assume that the data matrix $A \in \mathbf{R}^{m \times n}$ is columnwise centered, *i.e.*, the data points $a_1, \ldots, a_m \in \mathbf{R}^n$ whose transposes form the rows of A have zero mean.

(a) Let $P_k \in \mathbf{R}^{n \times k}$ be a matrix with orthonormal columns that represent the directions of some k-dimensional subspace in $\mathbf{R}^n$.

i. What is the total empirical variance captured by the projection onto this subspace, *i.e.*, the sum of the empirical variances of the projected data along each projected dimension?

ii. Formulate the problem of finding the k-dimensional subspace that maximizes the total empirical variance captured by the projection as an optimization problem as in (8.38). The final problem expression should not contain any norms.

(b) Let $A^TA = Q\Lambda Q^T$ be an eigenvalue decomposition of the matrix $A^TA \in \mathbf{S}^n_+$, where $Q \in \mathbf{R}^{n\times n}$ is an orthogonal matrix of eigenvectors and $\Lambda = \mathbf{diag}(\lambda_1, \ldots, \lambda_n) \in \mathbf{R}^{n\times n}$ is a diagonal matrix of eigenvalues with $\lambda_1 \geq \cdots \geq \lambda_n \geq 0$. For any matrix $P_k \in \mathbf{R}^{n\times k}$ with orthonormal columns, define

$$W = Q^T P_k \in \mathbf{R}^{n\times k},$$

and let w_i^T be the ith row of the matrix $W \in \mathbf{R}^{n\times k}$ for $i = 1, \ldots, n$.

i. Show that w_i satisfies $\sum_{i=1}^n \|w_i\|_2^2 = k$ and $0 \leq \|w_i\|_2^2 \leq 1$ for $i = 1, \ldots, n$. We will later use these properties to bound the optimal value of the problem you formulated in (a).

ii. Express the objective of the variance maximization problem you formulated in (a) in terms of the eigenvalues $\lambda_1, \ldots, \lambda_n$ of A^TA and $w_1, \ldots, w_n \in \mathbf{R}^k$.

iii. Show that the objective expression above is upper bounded by $\sum_{i=1}^k \lambda_i$, and this upper bound can be achieved by choosing

$$w_i = \begin{cases} e_i, & i = 1, \ldots, k, \\ 0, & i = k+1, \ldots, n. \end{cases}$$

This means that the optimal point of the variance maximization problem corresponds to taking the columns of P_k to be the first k eigenvectors of A^TA.

(c) Show that the optimal point of the low rank approximation problem (8.32) with $k > 1$ is obtained by projecting the data points onto the k-dimensional subspace spanned by the first k eigenvectors of A^TA.

8.4 *Equivalent formulations of quadratically regularized low rank approximation.* Consider the quadratically regularized low rank approximation problem

$$\text{minimize} \quad \|A - XY\|_F^2 + (\gamma/2)(\|X\|_F^2 + \|Y\|_F^2) \tag{8.58}$$

with variables $X \in \mathbf{R}^{m\times k}$ and $Y \in \mathbf{R}^{k\times n}$, where $\gamma > 0$ is the regularization parameter. This exercise shows that the problem (8.58) is equivalent to the rank constrained problem

$$\begin{array}{ll} \text{minimize} & \|A - Z\|_F^2 + \gamma\|Z\|_* \\ \text{subject to} & \mathbf{rank}\, Z \leq k, \end{array} \tag{8.59}$$

where $Z \in \mathbf{R}^{m\times n}$ is the optimization variable, and $\|\cdot\|_*$ is the nuclear norm of matrices, *i.e.*, the sum of the singular values of a matrix.

(a) First show that we have the following inequality: For any matrix $Z \in \mathbf{R}^{m\times n}$ with $\mathbf{rank}\, Z \leq k$, we have

$$\|Z\|_* \leq \frac{1}{2}(\|X\|_F^2 + \|Y\|_F^2),$$

where $X \in \mathbf{R}^{m\times k}$ and $Y \in \mathbf{R}^{k\times n}$ are any matrices such that $Z = XY$.
Hint. See §A.3.2 for some useful properties of the involved matrix norms.

(b) Using the result in (a), show that for any matrix $Z \in \mathbf{R}^{m\times n}$ with $\mathbf{rank}\, Z \leq k$, we have

$$\|Z\|_* = \inf_{XY=Z} \frac{1}{2}(\|X\|_F^2 + \|Y\|_F^2),$$

where the infimum is taken over all possible factorizations of Z into XY with $X \in \mathbf{R}^{m\times k}$ and $Y \in \mathbf{R}^{k\times n}$.

(c) Using the above results, conclude that the problems (8.58) and (8.59) are equivalent.

8.5 *Low rank approximation with Huber penalty.* Consider the generalized low rank approximation problem

$$\begin{array}{ll} \text{minimize} & \|S\|_{\text{sav}} + (1/2)\|N\|_F^2 + \gamma\|Z\|_* \\ \text{subject to} & S + N + Z = A \\ & \mathbf{rank}\, Z \leq k, \end{array} \tag{8.60}$$

where $S, N, Z \in \mathbf{R}^{m\times n}$ are the optimization variables, and $\|\cdot\|_{\text{sav}}$, $\|\cdot\|_F$, and $\|\cdot\|_*$ are the sum-absolute-value norm, Frobenius norm, and nuclear norm of matrices, respectively.

(a) What characteristics of the data matrix A are expected to be captured by the decomposition S, N, and Z in the problem (8.60), respectively?

(b) Show that the problem (8.60) is equivalent to

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^m \sum_{j=1}^n \phi(a_{ij} - z_{ij}) + \gamma\|Z\|_* \\ \text{subject to} & \mathbf{rank}\, Z \leq k, \end{array}$$

where $Z \in \mathbf{R}^{m\times n}$ is the variable, z_{ij} denotes the (i,j)th entry of Z, and $\phi(u)$ is the Huber penalty defined as

$$\phi(u) = \begin{cases} (1/2)u^2, & |u| \leq 1, \\ |u| - 1/2, & \text{otherwise.} \end{cases}$$

Hint. Consider the infimal convolution formulation of the Huber penalty obtained in exercise 4.1.

(c) Let $XY = Z$ be any factorization of $Z \in \mathbf{R}^{m\times n}$ with $X \in \mathbf{R}^{m\times k}$ and $Y \in \mathbf{R}^{k\times n}$. Show that the problem (8.60) is equivalent to

$$\text{minimize} \quad \sum_{i=1}^m \sum_{j=1}^n \phi(a_{ij} - x_i^T y_j) + (\gamma/2)(\|X\|_F^2 + \|Y\|_F^2)$$

with variables $X \in \mathbf{R}^{m\times k}$ and $Y \in \mathbf{R}^{k\times n}$, where x_i^T is the ith row of X and y_j is the jth column of Y, and $\phi(u)$ is the Huber penalty defined as above. You can directly use the results we obtained in exercise 8.4.

8.6 *Low rank approximation for abstract binary data.* Suppose we are given a data matrix $A \in \mathbf{R}^{m\times n}$, where the entries $a_{ij} \in \{-1, 1\}$ are binary for $i = 1, \ldots, m$ and $j = 1, \ldots, n$. Consider the low rank approximation problem

$$\text{minimize} \quad \sum_{i=1}^m \sum_{j=1}^n \phi(a_{ij}, x_i^T y_j) \tag{8.61}$$

with variables $X \in \mathbf{R}^{m\times k}$ and $Y \in \mathbf{R}^{k\times n}$, where x_i^T is the ith row of X and y_j is the jth column of Y. The function $\phi\colon \mathbf{R} \times \mathbf{R} \to \mathbf{R}$ is a cost function that measures the approximation error between the binary entry a_{ij} and the approximated value $x_i^T y_j$ for $i = 1, \ldots, m$ and $j = 1, \ldots, n$.

(a) Interpret the problem (8.61) when the cost function ϕ is given by

i. $\phi(a, z) = \log(1 + \exp(-az))$, and

ii. $\phi(a, z) = (1 - az)_+ = \max\{0, 1 - az\}$,

for $a \in \{-1, 1\}$ and $z \in \mathbf{R}$. In particular, for fixed $Y = \begin{bmatrix} y_1 & \cdots & y_n \end{bmatrix} \in \mathbf{R}^{k\times n}$, what does the problem (8.61) with the above cost functions reduce to?

(b) Explain why it is often necessary to include regularization terms, *e.g.*, a Frobenius norm regularizer, on the factor matrices X and Y in the problem (8.61) with the two cost functions in (a). What can go wrong without such regularization, and when?

Appendices

Appendix A

Mathematical background

A.1 Basic analysis

We start from some basic concepts and algebraic operations involving sets. A *set* is empty when it contains no elements, denoted as $\emptyset$. If a set contains only one element, then it is called a *singleton*. Let $\alpha \in \mathbf{R}$ be a real number, $x \in \mathbf{R}^n$ be a n-dimensional real vector, $T \in \mathbf{R}^{m\times n}$ be a real $m \times n$ matrix, and $A, B \subseteq \mathbf{R}^n$ be two subsets of $\mathbf{R}^n$. We define the following set operations:

- *Translation..* $x + A = \{x + a \mid a \in A\}$.
- *Addition.* $A + B = \{a + b \mid a \in A,\ b \in B\}$.
- *Scalar multiplication.* $\alpha A = \{\alpha a \mid a \in A\}$.
- *Matrix multiplication.* $TA = \{Ta \mid a \in A\}$.

In particular, the set addition $A + B$ defined above is also known as the *Minkowski sum* of sets A and B.

A.1.1 Supremum and infimum

Let $S \subseteq \mathbf{R}$ be a subset of real numbers. If there exists some $b \in \mathbf{R}$ such that $x \leq b$ for all $x \in S$, then S is *bounded above* and b is an *upper bound* of S. If there exists an upper bound b_0 of S such that $b_0 \leq b$ for all upper bounds b of S, then b_0 is called the *least upper bound*, or the *supremum* of S, denoted as

$$b_0 = \sup S.$$

If there is no upper bound of S, we write $\sup S = \infty$, and the set S is said to be *unbounded above*. If the set S is an empty set (*i.e.*, any $b \in \mathbf{R}$ is an upper bound of S), we define $\sup \emptyset = -\infty$. When $\sup S \in S$, we say the supremum is *attained*, or *achieved*.

Similarly, if there exists some $c \in \mathbf{R}$ such that $x \geq c$ for all $x \in S$, then S is *bounded below* and c is a *lower bound* of S. If there exists a lower bound c_0 of S

such that $c_0 \geq c$ for all lower bounds c of S, then c_0 is called the *greatest lower bound*, or the *infimum* of S, denoted as

$$c_0 = \inf S.$$

It is easy to see that

$$\inf S = -\sup(-S) = -\sup\{-x \mid x \in S\}.$$

If there is no lower bound of S, we write $\inf S = -\infty$, and the set S is said to be *unbounded below*. If the set S is an empty set (*i.e.*, any $c \in \mathbf{R}$ is a lower bound of S), we define $\inf \emptyset = \infty$. When $\inf S \in S$, we say the infimum is *attained*, or *achieved*.

If a set S is both bounded above and bounded below, we say it is *bounded*. If the set S is finite, then both $\sup S$ and $\inf S$ are attained, and they are simply the maximum and minimum of the elements in S, respectively.

A.1.2 Topology

The *Euclidean ball* with center $c \in \mathbf{R}^n$ and radius $r > 0$ is defined as

$$\mathcal{B}(c, r) = \{x \in \mathbf{R}^n \mid \|x - c\|_2 < r\},$$

where

$$\|x\|_2 = (x_1^2 + \cdots + x_n^2)^{1/2}$$

is the *Euclidean norm* of vector $x \in \mathbf{R}^n$ (see also §A.3.1).

Let $S \subseteq \mathbf{R}^n$ be a subset of an n-dimensional Euclidean space. The set S is said to be *open* if for every point $x \in S$, there exists some $\delta > 0$ such that the ball $\mathcal{B}(x, \delta)$ is contained in S, *i.e.*, $\mathcal{B}(x, r) \subseteq S$. The set S is said to be *closed* if its complement $S^c = \mathbf{R}^n \setminus S$ is open.

Let $C \subseteq \mathbf{R}^n$ be the *closure* of set S, which is defined as the intersection of all closed sets that contain S:

$$C = \bigcap \{E \subseteq \mathbf{R}^n \mid E \text{ closed and } S \subseteq E\},$$

i.e., the smallest closed set that contains S. Let $D \subseteq \mathbf{R}^n$ be the *interior* of set S, which is defined as

$$D = \{x \in S \mid \mathcal{B}(x, \delta) \subseteq S \text{ for some } \delta > 0\},$$

i.e., the largest open set contained in S. The set

$$B = C \setminus D \subseteq \mathbf{R}^n$$

is then the *boundary* of set S, which is the set of points that are in the closure of S but not in the interior of S.

A.2 Linear algebra

Given a matrix $A \in \mathbf{R}^{m\times n}$, the *range* (or *column space*) of A is defined as

$$\mathcal{R}(A) = \{Ax \mid x \in \mathbf{R}^n\} \subseteq \mathbf{R}^m,$$

which is the set of all vectors that can be expressed as a linear combination of the columns of A. The dimension of $\mathcal{R}(A)$ is called the *rank* of matrix A, denoted as $\mathbf{rank}\, A$, and is equal to the number of linearly independent columns of A. Hence, we have $\mathbf{rank}\, A \leq \min\{m, n\}$, and if $\mathbf{rank}\, A = \min\{m, n\}$, then A is said to have *full rank*. A closely related concept is the *nullspace* of A, defined as

$$\mathcal{N}(A) = \{x \in \mathbf{R}^n \mid Ax = 0\},$$

which is the set of all vectors that are mapped to the zero vector by A.

A.2.1 Spectral decomposition and definiteness

Given a matrix $A \in \mathbf{S}^n$, then there exists an *orthogonal* matrix $Q \in \mathbf{R}^{n\times n}$ (*i.e.*, $Q^TQ = I$) and a diagonal matrix $\Lambda = \mathbf{diag}(\lambda_1, \ldots, \lambda_n) \in \mathbf{R}^{n\times n}$ such that

$$A = Q\Lambda Q^T,$$

where $\lambda_1 \geq \cdots \geq \lambda_n$ are the *eigenvalues* of A, and the columns of Q are the corresponding *eigenvectors* of A. The *decomposition*, or *factorization*, of the matrix A as above is called the *spectral decomposition* (or *symmetric eigenvalue decomposition*) of A.

Usually, we denote the ith largest eigenvalue of $A \in \mathbf{S}^n$ as $\lambda_i(A)$. In particular, the maximum and minimum eigenvalues of A are denoted as $\lambda_{\max}(A) = \lambda_1(A)$ and $\lambda_{\min}(A) = \lambda_n(A)$, respectively, and satisfy

$$\lambda_{\max}(A) = \sup_{x\neq 0} \frac{x^TAx}{x^Tx} \qquad \text{and} \qquad \lambda_{\min}(A) = \inf_{x\neq 0} \frac{x^TAx}{x^Tx}.$$

A matrix $A \in \mathbf{S}^n$ is said to be *positive semidefinite* if for all $x \in \mathbf{R}^n$ and $x \neq 0$, we have

$$x^TAx \geq 0, \tag{A.1}$$

or equivalently, all eigenvalues of A are nonnegative, *i.e.*, $\lambda_{\min}(A) \geq 0$. This is denoted as $A \succeq 0$ or $A \in \mathbf{S}^n_+$. If strict inequality holds in (A.1) for all $x \neq 0$, then A is said to be *positive definite*, denoted as $A \succ 0$ or $A \in \mathbf{S}^n_{++}$, which is equivalent to all eigenvalues of A being positive, *i.e.*, $\lambda_{\min}(A) > 0$. Similarly, a matrix $A \in \mathbf{S}^n$ is said to be *negative semidefinite* if $-A$ is positive semidefinite, denoted as $A \preceq 0$, and *negative definite* if $-A$ is positive definite, denoted as $A \prec 0$.

A.2.2 Singular value decomposition

Given a matrix $A \in \mathbf{R}^{m\times n}$ with $\mathbf{rank}\, A = k$, then there exist orthogonal matrices $U \in \mathbf{R}^{m\times k}$ and $V \in \mathbf{R}^{n\times k}$, and a diagonal matrix $\Sigma = \mathbf{diag}(\sigma_1, \ldots, \sigma_k) \in \mathbf{R}^{k\times k}$, such that

$$A = U\Sigma V^T, \tag{A.2}$$

where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k > 0$ are the *singular values* of A. The columns of U and V are called the *left singular vectors* and *right singular vectors* of A, respectively. The decomposition of the matrix A as above is called the (compact) *singular value decomposition* (SVD) of A.

Full singular value decomposition

Suppose $A \in \mathbf{R}^{m\times n}$ with $\mathbf{rank}\, A = k$ has the SVD: $A = U_1\Sigma_1 V_1^T$ as in (A.2), where $U_1 \in \mathbf{R}^{m\times k}$, $V_1 \in \mathbf{R}^{n\times k}$, and $\Sigma_1 \in \mathbf{R}^{k\times k}$. Then there exist matrices $U_2 \in \mathbf{R}^{m\times(m-k)}$ and $V_2 \in \mathbf{R}^{n\times(n-k)}$ such that

$$U = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \in \mathbf{R}^{m\times m} \quad \text{and} \quad V = \begin{bmatrix} V_1 & V_2 \end{bmatrix} \in \mathbf{R}^{n\times n} \tag{A.3}$$

are orthogonal matrices. Define the matrix

$$\Sigma = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \in \mathbf{R}^{m\times n}. \tag{A.4}$$

Then we have the *full singular value decomposition* of A as

$$A = U\Sigma V^T,$$

where $U \in \mathbf{R}^{m\times m}$ and $V \in \mathbf{R}^{n\times n}$ are orthogonal matrices given by (A.3), and $\Sigma \in \mathbf{R}^{m\times n}$ is a diagonal matrix with the same singular values as A on its main diagonal as in (A.4).

Suppose $A \in \mathbf{R}^{m\times n}$ has the full SVD: $A = U\Sigma V^T$ as above, then we have

$$A^TA = V\Sigma U^TU\Sigma V^T = V\Sigma^2V^T, \tag{A.5}$$

where $V \in \mathbf{R}^{n\times n}$ is an orthogonal matrix and $\Sigma^2 \in \mathbf{R}^{n\times n}$ is a diagonal matrix with diagonal entries $\sigma_1^2, \ldots, \sigma_k^2$, and 0 (if any) on its main diagonal. The factorization (A.5) is the spectral decomposition of $A^TA \in \mathbf{S}^n$: The nonzero eigenvalues of A^TA are the squares of the nonzero singular values of A, and the columns of V are the corresponding eigenvectors of A^TA.

Usually, we denote the ith largest singular value of $A \in \mathbf{R}^{m\times n}$ (with $\mathbf{rank}\, A = k$) as $\sigma_i(A)$. In particular, the maximum and minimum singular values of A are denoted as $\sigma_{\max}(A)$ and $\sigma_{\min}(A)$, respectively, and satisfies

$$\sigma_{\max}(A) = \sup_{x\neq 0} \frac{\|Ax\|_2}{\|x\|_2} = \left(\lambda_{\max}(A^TA)\right)^{1/2}$$

and

$$\sigma_{\min}(A) = \begin{cases} \sigma_k(A), & k = \min\{m,n\} \\ 0, & k < \min\{m,n\}, \end{cases}$$

which is nonzero if and only if A has full rank.

A.2.3 Schur complement

Given a matrix $X \in \mathbf{S}^n$ that is partitioned as

$$X = \begin{bmatrix} A & B \\ B^T & C \end{bmatrix},$$

where $A \in \mathbf{S}^k$, $B \in \mathbf{R}^{k\times(n-k)}$, and $C \in \mathbf{S}^{n-k}$. If the matrix A is invertible, *i.e.*, $\det A \neq 0$, then the following matrix

$$S = C - B^T A^{-1} B \tag{A.6}$$

is called the *Schur complement* of A in X.

Determinant of block matrices

The determinant of X can be expressed in terms of the determinant of A and the Schur complement S as

$$\det X = \det A \det S = \det A \det(C - B^T A^{-1} B).$$

To show this, since $\det A \neq 0$, we could form the block lower triangular matrix

$$L = \begin{bmatrix} I & 0 \\ -B^T A^{-1} & I \end{bmatrix} \in \mathbf{R}^{n\times n},$$

whose determinant is $\det L = 1$, since all its diagonal entries are 1. Then we have

$$LX = \begin{bmatrix} I & 0 \\ -B^T A^{-1} & I \end{bmatrix} \begin{bmatrix} A & B \\ B^T & C \end{bmatrix} = \begin{bmatrix} A & B \\ 0 & C - B^T A^{-1} B \end{bmatrix}.$$

Now taking the determinant on both sides of the above equation, we have

$$\det X = \det L \det X = \det(LX) = \det A \det(C - B^T A^{-1} B) = \det A \det S,$$

where the third equality holds since LX is a block upper triangular matrix.

Definiteness conditions

Schur complement arises, for example, when we tries to perform partial minimization of a quadratic function over some of its variables. In particular, consider the following quadratic function $f\colon \mathbf{R}^k \times \mathbf{R}^{n-k} \to \mathbf{R}$ of $x \in \mathbf{R}^k$ and $y \in \mathbf{R}^{n-k}$:

$$f(x,y) = \begin{bmatrix} x \\ y \end{bmatrix}^T \begin{bmatrix} A & B \\ B^T & C \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = x^T A x + 2x^T B y + y^T C y.$$

Suppose the matrix $A \succ 0$ is positive definite, then the partial minimization problem

$$\text{minimize} \quad x^T A x + 2y^T B^T x + y^T C y \tag{A.7}$$

with variable $x \in \mathbf{R}^k$ is a convex quadratic optimization problem. Hence, a solution to the problem (A.7) can be obtained by setting the gradient of f with respect to the variable x to zero, *i.e.*,

$$\nabla_x f(x, y) = 2Ax + 2By = 0,$$

which gives the solution $x = -A^{-1}By$. The optimal value of the problem (A.7) is then given by

$$\begin{aligned}
\inf_{x \in \mathbf{R}^k} f(x, y) &= \inf_{x \in \mathbf{R}^k} \begin{bmatrix} x \\ y \end{bmatrix}^T \begin{bmatrix} A & B \\ B^T & C \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \\
&= (-A^{-1}By)^T A(-A^{-1}By) + 2(-A^{-1}By)^T By + y^T Cy \\
&= y^T B^T A^{-1} A A^{-1} By - 2y^T B^T A^{-1} By + y^T Cy \\
&= y^T (C - B^T A^{-1} B) y \\
&= y^T S y,
\end{aligned}$$

where S is the Schur complement of A in X, given by (A.6). We have several important implications of the above result:

- If $A \succ 0$, then $X \succeq 0$ if and only if $S \succeq 0$.
- The matrix $X \succ 0$ if and only if $A \succ 0$ and $S \succ 0$.

These conditions are sometimes called the *Schur complement definiteness conditions* for symmetric matrices. They are potentially useful, for example, for transforming certain (matrix) inequalities involving matrix inverses into linear matrix inequalities. (See, *e.g.*, exercise 6.1.)

A.3 Norms

A.3.1 Inner products

The *standard inner product* of two vectors $x, y \in \mathbf{R}^n$ is defined as

$$x^T y = x_1 y_1 + \cdots + x_n y_n. \tag{A.8}$$

The *norm* associated with the standard inner product on $\mathbf{R}^n$ is the *Euclidean norm*, or the *ℓ_2-norm*, defined as

$$\|x\|_2 = (x^T x)^{1/2} = (x_1^2 + \cdots + x_n^2)^{1/2}.$$

One of the most important relationship between the standard inner product (A.8) and the associated Euclidean norm is the *Cauchy-Schwarz inequality*, which states that we always have the following inequality

$$|x^T y| \leq \|x\|_2 \|y\|_2$$

holds for all $x, y \in \mathbf{R}^n$, and the equality is achieved if and only if x and y are linearly dependent, *i.e.*, $x = \alpha y$ for some $\alpha \in \mathbf{R}$.

The standard inner product can be generalized to other vector spaces. For example, the standard inner product on the space of $m \times n$ real matrices $\mathbf{R}^{m\times n}$ is defined as

$$\mathbf{tr}(X^T Y) = \sum_{i=1}^{m}\sum_{j=1}^{n} X_{ij}Y_{ij}, \tag{A.9}$$

for $X, Y \in \mathbf{R}^{m\times n}$, where $\mathbf{tr}$ denotes the *trace* of a (square) matrix, *i.e.*, the sum of its diagonal entries. Note that the inner product for matrices given by (A.9) can be interpreted as the standard vector inner product (A.8) on the vectors in $\mathbf{R}^{mn}$ obtained by stacking the columns (or rows) of the matrices into vectors. The associated norm of (A.9) is the *Frobenius norm*, defined as

$$\|X\|_F = \left(\mathbf{tr}(X^T X)\right)^{1/2} = \left(\sum_{i=1}^{m}\sum_{j=1}^{n} X_{ij}^2\right)^{1/2}.$$

Similarly, the Frobenius norm can be viewed as the Euclidean norm of the vector obtained by stacking the columns (or rows) of the matrix X into a vector in $\mathbf{R}^{mn}$.

Remark A.1 Unlike vectors, the ℓ_2-norm of a matrix in $\mathbf{R}^{m\times n}$ is a different norm from the Frobenius norm: For $X \in \mathbf{R}^{m\times n}$, the ℓ_2-norm is defined as the maximum singular value of X, *i.e.*,

$$\|X\|_2 = \sigma_{\max}(X),$$

and is also known as the *spectral norm*; see example A.3. When $n = 1$, the matrix $X \in \mathbf{R}^{m\times 1}$ reduces to a vector, and the spectral norm coincides with the Euclidean norm.

As a special case of (A.9), the standard inner product on the space of $n \times n$ symmetric matrices $\mathbf{S}^n$ is given by

$$\mathbf{tr}(XY) = \sum_{i=1}^{n}\sum_{j=1}^{n} X_{ij}Y_{ij} = \sum_{i=1}^{n} X_{ii}Y_{ii} + 2\sum_{i=1}^{n}\sum_{j=i+1}^{n} X_{ij}Y_{ij},$$

for $X, Y \in \mathbf{S}^n$.

Unitary invariance of the Frobenius norm

One important property of the Frobenius norm is that it is *unitarily invariant*, meaning that for any $X \in \mathbf{R}^{m\times n}$ and any orthogonal matrices $U \in \mathbf{R}^{m\times m}$ and $V \in \mathbf{R}^{n\times n}$, we have the following equality:

$$\|UXV^T\|_F = \|X\|_F.$$

In other words, the Frobenius norm of a matrix X is invariant under orthogonal transformations of the matrix from both sides.

To prove this property, according to the definition of the Frobenius norm, we have

$$\begin{aligned}\|UXV^T\|_F^2 &= \mathbf{tr}\left((UXV^T)^T UXV^T\right) = \mathbf{tr}(VX^TU^TUXV^T)\\ &= \mathbf{tr}(VX^TXV^T) = \mathbf{tr}(X^TXV^TV)\\ &= \mathbf{tr}(X^TX) = \|X\|_F^2.\end{aligned}$$

The first equality follows from the definition of the Frobenius norm, the second equality follows from the fact that U is an orthogonal matrix, so $U^TU = I$. The third equality follows from the cyclic property of the trace function, which states that for any matrices A, B, C such that the products are well-defined, we have $\mathbf{tr}(ABC) = \mathbf{tr}(BCA) = \mathbf{tr}(CAB)$, and the fourth equality follows again from the orthogonality of V.

A.3.2 Vector and matrix norms

A function $f\colon \mathbf{R}^n \to \mathbf{R}$ with $\mathbf{dom}\, f = \mathbf{R}^n$ is called a *norm* if it satisfies the following properties:

- *Nonnegativity.* For all $x \in \mathbf{R}^n$, $f(x) \geq 0$.
- *Definiteness.* For all $x \in \mathbf{R}^n$, $f(x) = 0$ if and only if $x = 0$.
- *Homogeneity.* For all $x \in \mathbf{R}^n$ and $\alpha \in \mathbf{R}$, $f(\alpha x) = |\alpha| f(x)$.
- *Triangle inequality.* For all $x, y \in \mathbf{R}^n$, $f(x + y) \leq f(x) + f(y)$.

As a simple example, the absolute value function $f(x) = |x|$ is a norm on $\mathbf{R}$. Geometrically, the absolute value function measures the *distance* between two real numbers $x, y \in \mathbf{R}$ as $|x - y|$. As a generalization, for $x, y \in \mathbf{R}^n$ and f being a norm, the quantity $f(x - y)$ also defines *some* distance between points x and y, where the notion of distance depends on the specific choice of the norm. Hence, we usually denote a norm function f as $\|\cdot\|$, suggesting that it is some generalization of the absolute value function on real numbers, and use the notation

$$\mathbf{dist}(x, y) = \|x - y\|$$

to denote the distance between two points $x, y \in \mathbf{R}^n$ in the norm $\|\cdot\|$. Using this notation, the distance between two *sets of points* $C, D \subseteq \mathbf{R}^n$, in the norm $\|\cdot\|$, is defined as

$$\mathbf{dist}(C, D) = \inf\{\|x - y\| \mid x \in C, y \in D\}.$$

Note that whenever the norm notation $\|\cdot\|$ appears without subscripts, it can refer to any norm; if we want to talk about a specific norm, we use subscripts to indicate which norm we are referring to.

Every norm $\|\cdot\|$ is associated with a *norm ball*, defined as

$$B(x_0, r) = \{x \in \mathbf{R}^n \mid \|x - x_0\| \leq r\},$$

where $x_0 \in \mathbf{R}^n$ is the center and $r > 0$ is the radius. If $x_0 = 0$ and $r = 1$, the norm ball $B(0, 1)$ is also called the *unit ball* of the norm.

Example A.1 *Vector norms.* Many commonly used vector norms are special cases of the ℓ_p*-norm*, which is defined as

$$\|x\|_p = (|x_1|^p + |x_2|^p + \cdots + |x_n|^p)^{1/p}$$

for $p \geq 1$ and $x \in \mathbf{R}^n$. Some special cases include:

- $p = 1$: The ℓ_1*-norm*, or *Manhattan norm*, is given by

  $$\|x\|_1 = |x_1| + |x_2| + \cdots + |x_n|,$$

  which is the sum of the absolute values of all entries in x.
- $p = 2$: The ℓ_2*-norm*, or *Euclidean norm*, is given by

  $$\|x\|_2 = (x_1^2 + x_2^2 + \cdots + x_n^2)^{1/2}.$$

- $p \to \infty$: The ℓ_∞*-norm*, or *Chebyshev norm*, is given by

  $$\|x\|_\infty = \lim_{p\to\infty} \|x\|_p = \max\{|x_1|, |x_2|, \ldots, |x_n|\},$$

  which is the maximum absolute value among all entries in x.

The unit balls of these norms on $\mathbf{R}^2$ are illustrated in figure 2.9, page 31.

Example A.2 *Componentwise matrix norms.* The ℓ_1-, ℓ_2-, and ℓ_∞-norms for vectors can be generalized to some matrix $X \in \mathbf{R}^{m\times n}$ by applying them to the vector in $\mathbf{R}^{mn}$ formed by stacking all entries of the matrix, which gives the following *componentwise* matrix norms:

- *Sum-absolute-value norm*:

  $$\|X\|_{\text{sav}} = \sum_{i=1}^{m}\sum_{j=1}^{n} |X_{ij}|.$$

- *Frobenius norm*:

  $$\|X\|_F = (\mathbf{tr}(X^T X))^{1/2} = \left(\sum_{i=1}^{m}\sum_{j=1}^{n} X_{ij}^2\right)^{1/2}.$$

- *Max-absolute-value norm*:

  $$\|X\|_{\text{mav}} = \max\{|X_{ij}| \mid i = 1, \ldots, m,\ j = 1, \ldots, n\}.$$

As we have seen in remark A.1 for the Frobenius norm, these componentwise matrix norms are different from the matrix ℓ_1-, ℓ_2-, and ℓ_∞-norms; see example A.3.

Induced matrix norms

Let $\|\cdot\|$ be a norm defined both on $\mathbf{R}^m$ and $\mathbf{R}^n$. The *matrix norm* of $A \in \mathbf{R}^{m\times n}$ induced by the vector norm $\|\cdot\|$ is defined as

$$\|A\| = \sup\{\|Ax\| \mid \|x\| \leq 1\} = \sup_{x\neq 0} \frac{\|Ax\|}{\|x\|}.$$

This is also called the *operator norm* of matrix A induced by $\|\cdot\|$.

Example A.3 *Matrix norms.* Suppose $A \in \mathbf{R}^{m\times n}$. As a simple example, the matrix ℓ_2-norm, or *spectral norm*, is given by

$$\|A\|_2 = \sup\{\|Ax\|_2 \mid \|x\|_2 \leq 1\} = \sigma_{\max}(A),$$

where $\sigma_{\max}(A)$ denotes the maximum singular value of matrix A. Recalling that the maximum eigenvalue of A^TA is given by

$$\lambda_{\max}(A^TA) = \sup_{x\neq 0} \frac{x^T(A^TA)x}{x^Tx} = \sup_{x\neq 0} \frac{(Ax)^T(Ax)}{x^Tx} = \left(\sup_{x\neq 0} \frac{\|Ax\|_2}{\|x\|_2}\right)^2 = \|A\|_2^2,$$

we have

$$\sigma_{\max}(A) = \left(\lambda_{\max}(A^TA)\right)^{1/2}.$$

As another example, the matrix ℓ_1-norm is given by

$$\|A\|_1 = \sup\{\|Ax\|_1 \mid \|x\|_1 \leq 1\} = \max_{j=1,\ldots,n} \sum_{i=1}^{m} |A_{ij}|,$$

which is the maximum absolute column sum of matrix A.

Finally, the matrix ℓ_∞-norm is given by

$$\|A\|_\infty = \sup\{\|Ax\|_\infty \mid \|x\|_\infty \leq 1\} = \max_{i=1,\ldots,m} \sum_{j=1}^{n} |A_{ij}|,$$

which is the maximum absolute row sum of matrix A.

Dual norm

Let $\|\cdot\|$ be a norm defined on $\mathbf{R}^n$. The *dual norm* of $\|\cdot\|$ for $x \in \mathbf{R}^n$ is defined as

$$\|x\|_{\text{dual}} = \sup\{x^Ty \mid \|y\| \leq 1\}.$$

The dual of the ℓ_p-norm is the ℓ_q-norm, where $1/p + 1/q = 1$. In particular, we have the following special cases:

- The dual norm of the ℓ_1-norm is the ℓ_∞-norm.
- The dual norm of the ℓ_2-norm is itself.
- The dual norm of the ℓ_∞-norm is the ℓ_1-norm.

Another useful matrix norm in practice is the *nuclear norm*, which is the dual norm of the spectral norm:

$$\|X\|_* = \sup\{\mathbf{tr}(X^T Y) \mid \|Y\|_2 \leq 1\} = \sum_{i=1}^{\min\{m,n\}} \sigma_i(X),$$

where $\sigma_i(X)$ denotes the ith singular value (including zeros) of matrix $X \in \mathbf{R}^{m\times n}$. An application involving these norms can be found in example 2.11.

Remark A.2 An equivalent definition of the dual norm is given by adding the absolute value inside the supremum, *i.e.*,

$$\|x\|_{\text{dual}} = \sup\{|x^T y| \mid \|y\| \leq 1\},$$

which also appears commonly in textbooks. To see the equivalence, let $x \in \mathbf{R}^n$ be given. We first note that

$$\sup_{\|y\|\leq 1} |x^T y| \geq \sup_{\|y\|\leq 1} x^T y,$$

since $|x^T y| \geq x^T y$ for all $y \in \mathbf{R}^n$. On the other hand, for any $y \in \mathbf{R}^n$ with $\|y\| \leq 1$, we have $-y \in \mathbf{R}^n$ with $\|-y\| \leq 1$, and hence

$$|x^T y| = \max\{x^T y, x^T(-y)\} \leq \sup_{\|y\|\leq 1} x^T y,$$

so we have

$$\sup_{\|y\|\leq 1} |x^T y| \leq \sup_{\|y\|\leq 1} x^T y.$$

Hence, we have conclude that $\sup_{\|y\|\leq 1} |x^T y| = \sup_{\|y\|\leq 1} x^T y$, and the two definitions of the dual norm are equivalent.

Bibliographical notes

The standard reference for analysis is the book by Rudin [Rud76]. Some recent books, *e.g.*, by Abbott [Abb15] and Tao [Tao22], also include useful material on this topic.

For linear algebra, we refer readers to the books by Strang [Str06] and Meyer [Mey23]. More application-oriented material on linear algebra and examples can be found in the book by Boyd and Vandenberghe [BV18]. Horn and Johnson [HJ12] includes more advanced topics on matrix analysis.

The original idea of Schur complement matrix goes back to the 1850s in the work by Sylvester [Syl51], and was later used by Issai Schur to prove *Schur's lemma* [Sch17] in the 1910s. The name *Schur complement* was first used by Haynsworth [Hay68] in 1968. It is also sometimes referred to as the *Feshbach map* in physics [Fes58]. Boyd and Vandenberghe [BV04, A5.5] provides a short review of Schur complement. See Zhang [Zha05] for a comprehensive review about its full history and some applications.

Norms and inner products are discussed in many textbooks on functional analysis, include Yosida [Yos12] and Muscat [Mus24]. Boyd and Vandenberghe [BV04, §A.1] and the listed references also provide a helpful review of norm-related material used in optimization.

We do not mention functions and derivatives in this appendix. These materials can be found in standard textbooks on analysis or multivariable calculus, *e.g.*, by Apostol [Apo67] and Marsden and Tromba [MT11]. Materials on these topics from a optimization practitioner perspective can be found in the appendix of [BV04, §A.3 and §A.4], and [BV18, §C.1 and §C.2].

Appendix B

Disciplined convex analysis and programming

B.1 Standard convexity verification

We first consider the problem of verifying the convexity of a given function $f\colon \mathbf{R}^n \to \mathbf{R}$ with $\mathbf{dom}\, f \subseteq \mathbf{R}^n$ (potentially by a human or a computer). There are several analytical approaches to address this problem.

Jensen's inequality

The most direct approach is to verify the definition of convexity using Jensen's inequality, *i.e.*, for all $x, y \in \mathbf{dom}\, f$ and $\theta \in [0,1]$, check whether

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta) f(y). \tag{B.1}$$

It is easy to see that this approach is equivalent to checking whether for all $x \in \mathbf{dom}\, f$ and $v \in \mathbf{R}^n$, the function

$$g(t) = f(x + tv)$$

is convex on its domain $\{t \in \mathbf{R}^n \mid x + tv \in \mathbf{dom}\, f\}$. In other words, the functions f is convex if and only if it is convex when restricted to any line that intersects its domain. This approach compared to directly verifying Jensen's inequality is often more convenient, since now we only need to verify the convexity of a function with one variable. The following example illustrates this idea.

Example B.1 *Log-determinant function.* Consider the log-determinant function defined as $f(X) = \log\det X$ with $\mathbf{dom}\, f = \mathbf{S}^n_{++}$. We can verify its concavity by restricting f to an arbitrary line $X = Z + tV$, where $Z \in \mathbf{S}^n_{++}$ and $V \in \mathbf{S}^n$. Defining the function

$$g(t) = f(Z + tV) = \log\det(Z + tV)$$

with domain $\mathbf{dom}\, g = \{t \in \mathbf{R} \mid Z + tV \succ 0\}$, then we have

$$\begin{aligned} g(t) &= \log\det(Z + tV) \\ &= \log\det(Z^{1/2}(I + tZ^{-1/2}VZ^{-1/2})Z^{1/2}) \\ &= \log\det(Z) + \log\det(I + tZ^{-1/2}VZ^{-1/2}) \\ &= \log\det(Z) + \sum_{i=1}^{n} \log(1 + t\lambda_i), \end{aligned}$$

where $\lambda_1, \ldots, \lambda_n$ are the eigenvalues of the matrix $Z^{-1/2}VZ^{-1/2}$. Hence, we have

$$g'(t) = \sum_{i=1}^{n} \frac{\lambda_i}{1 + t\lambda_i} \quad \text{and} \quad g''(t) = -\sum_{i=1}^{n} \frac{\lambda_i^2}{(1 + t\lambda_i)^2} \leq 0,$$

which shows that g is concave for all t in its domain, and thus the log-determinant function f is concave.

Second-order conditions

If the function f is twice differentiable on its domain $\mathbf{dom}\, f$, then we can verify its convexity by checking whether its Hessian matrix $\nabla^2 f(x)$ is positive semidefinite for all $x \in \mathbf{dom}\, f$. We have seen in §2.3.3 that this approach can be used to verify the convexity of many basic functions. Here we present another more complicated example.

Example B.2 *Log-sum-exp function.* Consider the log-sum-exp function, which is defined as $f(x) = \log \sum_{i=1}^{n} \exp x_i$ with $\mathbf{dom}\, f = \mathbf{R}^n$. We can verify its convexity by computing its Hessian matrix. Let $z = (e^{x_1}, \ldots, e^{x_n})$, then the gradient of f is

$$\nabla f(x) = \frac{1}{\mathbf{1}^T z} z,$$

so its Hessian can be expressed as

$$\nabla^2 f(x) = \frac{1}{(\mathbf{1}^T z)^2} \left((\mathbf{1}^T z)\, \mathbf{diag}(z) - zz^T \right).$$

To verify that $\nabla^2 f(x) \succeq 0$, let $v \in \mathbf{R}^n$ be an arbitrary vector, then we have

$$\begin{aligned} v^T \nabla^2 f(x) v &= \frac{1}{(\mathbf{1}^T z)^2} \left((\mathbf{1}^T z) \sum_{i=1}^{n} z_i v_i^2 - \left(\sum_{i=1}^{n} z_i v_i \right)^2 \right) \\ &= \frac{1}{(\mathbf{1}^T z)^2} \left(\left(\sum_{i=1}^{n} z_i \right) \left(\sum_{i=1}^{n} z_i v_i^2 \right) - \left(\sum_{i=1}^{n} z_i v_i \right)^2 \right). \end{aligned}$$

Let $a = (\sqrt{z_1}, \ldots, \sqrt{z_n})$ and $b = (v_1\sqrt{z_1}, \ldots, v_n\sqrt{z_n})$, then by the Cauchy-Schwarz inequality, we have

$$(a^T a)(b^T b) = \left(\sum_{i=1}^{n} z_i \right) \left(\sum_{i=1}^{n} z_i v_i^2 \right) \geq (a^T b)^2 = \left(\sum_{i=1}^{n} z_i v_i \right)^2,$$

which shows that $v^T \nabla^2 f(x) v \geq 0$ for all $v \in \mathbf{R}^n$, and thus the log-sum-exp function f is convex.

Automatic convexity verification

As we may see from the examples B.1 and B.2 that verifying the convexity of some function directly from the definition (B.1) or second-order conditions can be tedious and complicated. In particular, a human needs to be well trained and experienced in convex analysis to carry out these procedures successfully. Even for a computer, these procedures may involve complicated symbolic computations that are not easy to implement. If we were to implement some computer algorithm based on these two approaches for automatic convexity verification, probably the most practical scenario is to disprove the convexity of a function, *i.e.*, to find the counter examples when the function is not convex.

For example, suppose we want to verify whether a given function $f : \mathbf{R}^n \to \mathbf{R}$ is convex, then we can randomly sample a large number of points $x, y \in \mathbf{dom}\, f$ and $\theta \in [0, 1]$, and check whether Jensen's inequality (B.1) holds for these points. If we find any violation of the inequality, then we can conclude that the function is not convex. Similarly, if the function is twice differentiable, we can randomly sample a large number of points $x \in \mathbf{dom}\, f$, and evaluate whether the Hessian matrices $\nabla^2 f(x)$ at these points are all positive semidefinite. If we find any point where the Hessian is not positive semidefinite, then we can conclude that the function is not convex. However, the converse is not true: Even if we do not find any violation after a large number of samples, we still have no information regarding the convexity of the function.

B.2 Constructive convex analysis

To lower the expert barrier for humans and computers to verify the convexity of functions, we can make use of the convexity preserving functional operations presented in §2.4, where the basic idea is to show that the target function can be obtained from simple convex functions (such as those presented in §2.3.3) by a sequence of operations that preserve convexity. In other words, the target function f is carefully parsed into a *composition tree*, where the nodes represent some functions with known convexity, or some convexity preserving operations. Then to verify the convexity of f, we only need to traverse the composition tree from the leaves to the root, and at each node, check whether the convexity is preserved. This *constructive* convexity verification procedure only requires us to know a library of basic *atomic* convex functions as well as a set of functional operations that preserve convexity, and it is expected to be easily implemented on computers as an automatic procedure.

Example B.3 Consider the function $f : \mathbf{R} \to \mathbf{R}$ given by $f(x) = 2x^2 + 3$. The composition tree for this function is shown in figure B.1. Starting from the leaves, we see that the variable x is affine. Then the node above it represents the function $(\,\cdot\,)^2$, which is convex and nonnegative, so by the composition rule for convex functions, the

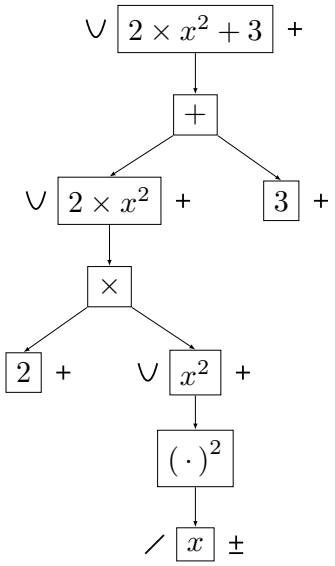

Figure B.1 Composition tree for the function $f(x) = 2x^2 + 3$. The curvature and sign of the involved functions are indicated by the symbols on the left and right of the node, respectively.

function x^2 is convex and nonnegative. Next, multiplying it by the positive constant 2 preserves convexity and sign, so the function $2x^2$ is convex and nonnegative. The constant function 3 is nonnegative, so their sum $2x^2 + 3$ is convex and nonnegative as well.

Example B.4 We should note that how the target function is parsed into a composition tree is not unique, and it may affect the success of the convexity verification. For example, consider the function $f\colon \mathbf{R} \to \mathbf{R}$ given by $f(x) = \sqrt{1 + x^2}$. Two possible composition trees for this function are shown in figure B.2. In the left tree, starting from the leaves, we see that the variable x is affine, then by the composition rule, the function x^2 is convex and nonnegative. Next, adding 1 (which is a nonnegative constant) preserves convexity and sign, so the function $1 + x^2$ is convex and nonnegative. However, since the function $\sqrt{\cdot}$ is concave and nondecreasing on $\mathbf{R}_+$, by the composition rule for concave functions, we do not know the curvature of $\sqrt{1 + x^2}$, so the constructive convex analysis fail here.

In fact, the function $f(x) = \sqrt{1 + x^2}$ is a convex function of x, and constructive convex analysis will indeed verify it as convex if we parse the function into the right tree. In this case, since the variable x is affine and the function that maps $x \mapsto (1, x)$ is also affine, we conclude that the function $(1, x)$ is affine (in the variable x). Next, the function $\|\cdot\|_2$ is convex and nonnegative, so by the composition rule for convex functions, the function $\|(1, x)\|_2 = \sqrt{1 + x^2}$ is convex and nonnegative.

On the other hand, we should note that this approach of convexity verification is not complete in the sense that there exist convex functions that cannot be con-

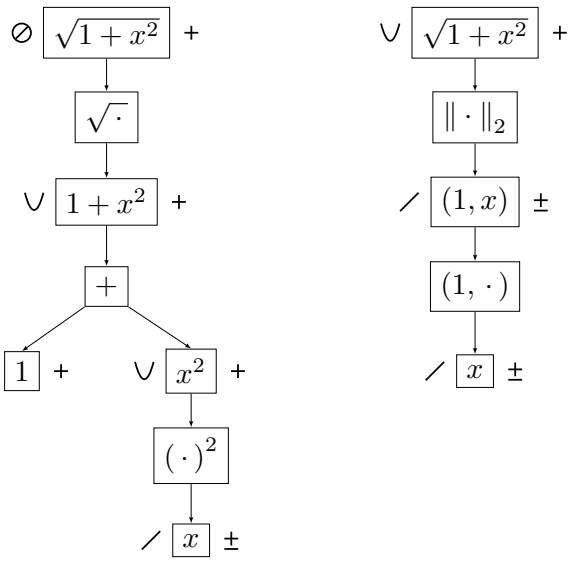

Figure B.2 Two possible composition trees for the function $f(x) = \sqrt{1+x^2}$. The curvature and sign of the involved functions are indicated by the symbols on the left and right of the node, respectively.

structed this way, especially when the library of atomic convex functions is small. For example, we have shown above that the log-sum-exp function is convex using the second-order condition, but it is not obvious how to construct this function using basic convex functions and functional operations, even if it only involves a few compositions on the logarithm and exponential functions. Nevertheless, if necessary, we can always enlarge the atomic functions library to include such functions once they appear, so that they can be directly used for the convexity verification of other more complicated functions.

B.3 Disciplined convex programming

Now we extend the ideas of constructive convex analysis to optimization problems, and consider the inverse problem of convexity verification, *i.e.*, how to specify an optimization problem in a way that can be automatically verified as convex. This leads to the framework of *disciplined convex programming* (DCP). The fundamental philosophy of DCP is, rather than simply construct the objective function and constraints without advance regard for convexity, we draw from a library of atomic functions whose convexity properties are already known, and combine them in ways which constructive convex analysis insures will produce convex results.

A DCP problem has the specific form

$$\begin{array}{ll} \text{minimize/maximize} & f(x) \\ \text{subject to} & p_i(x) \sim q_i(x), \quad i = 1, \ldots, m, \end{array} \tag{B.2}$$

where $x \in \mathbf{R}^n$ is the variable, the function $f\colon \mathbf{R}^n \to \mathbf{R}$ is the objective, and $p_i, q_i\colon \mathbf{R}^n \to \mathbf{R}$ are the left-hand side and right-hand side functions of the ith constraint. The relational operator $\sim$ represents one of the relations $\leq$, $\geq$, or $=$. In DCP this problem must be convex, which imposes additional rules on the curvature of the involved functions in the problem (B.2), which are listed below.

- For a minimization problem, the objective function f must be convex; for a maximization problem, f must be concave.
- For a constraint of the form $p_i(x) \leq q_i(x)$, the function p_i must be convex, and q_i must be concave.
- For a constraint of the form $p_i(x) \geq q_i(x)$, the function p_i must be concave, and q_i must be convex.
- For a constraint of the form $p_i(x) = q_i(x)$, both p_i and q_i must be affine.

Note that functions that are affine (and of course constant), *i.e.*, are both convex and concave, can match either curvature requirement. For example, we can minimize or maximize an affine function, and an affine function can appear on either side of an inequality constraint. Besides, these rules can be used to represent feasibility problems as well, by simply defining the objective function as a constant function, *e.g.*, $f(x) = 0$.

To verify the convexity of a DCP problem in the form (B.2), we can similarly parse *the problem* into a composition tree. Then, we apply the constructive convex analysis procedure to verify the convexity of the objective function and the constraints one by one, and finally check whether these involved functions satisfy the curvature requirements listed above. The following example illustrates these ideas.

Example B.5 We consider the following optimization problem:

$$\begin{array}{ll} \text{minimize} & cy \\ \text{subject to} & \exp(y) \leq \log(a\sqrt{x} + b) \\ & ax + by = d, \end{array} \tag{B.3}$$

where the variable are $x \in \mathbf{R}_+$, $y \in \mathbf{R}$, and the constants $a, b, c, d > 0$ are given problem data. The composition trees for the involved functions are shown in figure B.3.

We first verify the convexity of the involved functions.

- *Objective function.* Since the variable $y \in \mathbf{R}$ is affine, multiplying it by the positive constant c preserves convexity, so the objective function cy is affine.
- *Left-hand side of the inequality constraint.* Since the variable y is affine, and the function $\exp(\,\cdot\,)$ is convex and nonnegative, the function $\exp(y)$ is convex and nonnegative.
- *Right-hand side of the inequality constraint.* Since the variable $x \in \mathbf{R}_+$ is affine, and the square root function $\sqrt{\cdot}$ is concave and nonnegative on $\mathbf{R}_+$, the function $\sqrt{x}$ is concave and nonnegative. Then multiplying it by the positive constant a and adding the nonnegative constant b preserves concavity and sign, so the function $a\sqrt{x}+b$ is concave and nonnegative. Finally, since the function $\log(\,\cdot\,)$ is concave and nondecreasing on $\mathbf{R}_+$, by the composition rule for concave functions, the function $\log(a\sqrt{x} + b)$ is concave.

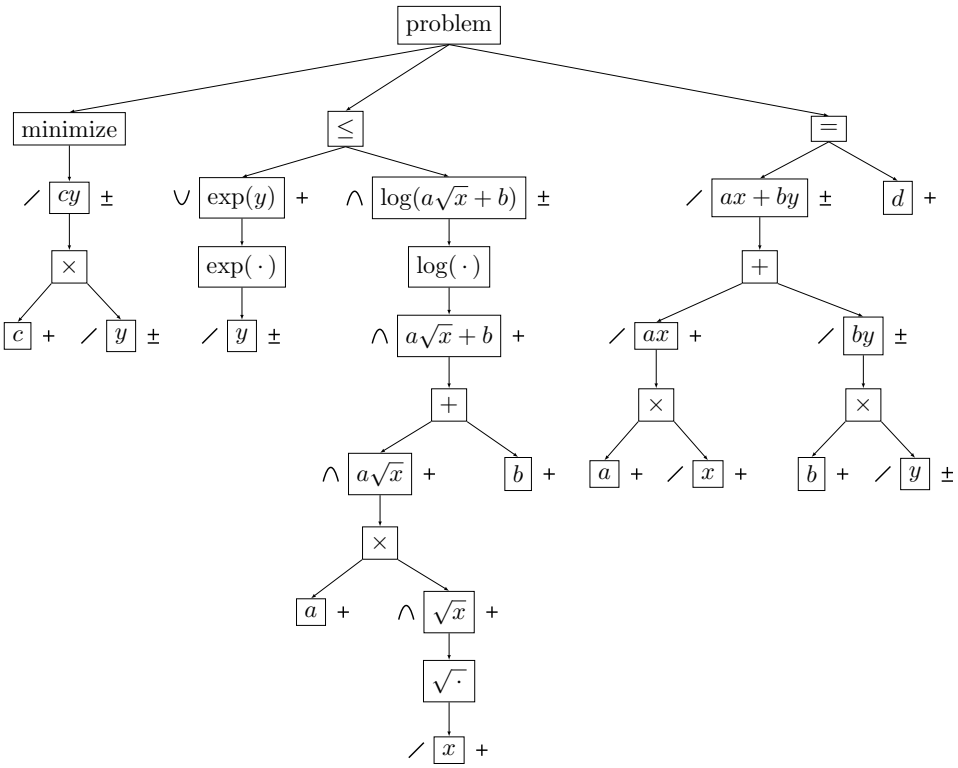

Figure B.3 Composition tree of the problem (B.3). The curvature and sign of the involved functions are indicated by the symbols on the left and right of the node, respectively.

- *Equality constraint.* The right-hand side is a constant function d, which is (of course) affine. On the left-hand side, since the variables x and y are both affine, multiplying them by the positive constants a and b leads to the affine functions ax and by. Finally, the function $ax + by$ is the addition of two affine functions and is hence affine.

Now we check whether these functions satisfy the DCP curvature requirements of the objective and constraints.

- *Objective function.* Since the problem is a minimization problem, the objective function cy must be convex, which holds since it is affine.
- *Inequality constraint.* For the constraint $\exp(y) \leq \log(a\sqrt{x}+b)$, the left-hand side $\exp(y)$ must be convex, and the right-hand side $\log(a\sqrt{x} + b)$ must be concave, which both hold.
- *Equality constraint.* For the constraint $ax + by = d$, both sides must be affine, which is also satisfied.

Since all the involved functions satisfy the DCP curvature requirements, we conclude that the problem (B.3) is convex.

Bibliographical notes

The idea of DCP was initially described in Grant *et al.* [GBY06] and in Grant's PhD thesis [Gra05] as part of the CVX [GB14] modeling system (or *domain specific language*) for MATLAB. It has been later implemented by several other modeling systems in different programming languages, including Convex.jl [UMZ+14] for Julia, CVXPY [DB16, AVDB18] for Python, and CVXR [FNB20] for R.

As the name suggests, DCP was initially designed for modeling convex optimization problems, but has been extended over the years to handle *stochastic* [AKDB15], *convex-concave* [SDGB16], *multi-convex* [SDU+17], *geometric* [ADB19], *quasiconvex* [AB20], *saddle* [SLB24], and *biconvex* [ZB25] optimization problems as well.

Some other related modeling frameworks for different types of mathematical optimization problems include NCVX [DTB17] for modeling and solving problems with convex objectives and variables from a nonconvex set; SnapVX [HWD+17] for convex optimization problems defined on graphs; GPkit [BDH20] for defining and manipulating geometric programming models; PICOS [SS22] as a Python interface for conic and mixed integer optimization problems; and OSMM [SAB23] for minimizing oracle-structured composite convex functions. Just to mention a few.

Appendix C

Technical issues in sequential methods

C.1 Alternate convex search

C.1.1 Proximal regularization

Consider using the ACS procedure (algorithm 3.1) to approximately solve the biconvex optimization problem

$$\begin{array}{lll} \text{minimize} & f_0(x,y) & \\ \text{subject to} & f_i(x,y) \leq 0, & i=1,\ldots,m \\ & h_i(x,y)=0, & i=1,\ldots,p \end{array} \tag{C.1}$$

with variables $x \in \mathbf{R}^n$ and $y \in \mathbf{R}^k$. At the $(k+1)$th iteration, the variables are updated according to

$$\begin{array}{rcll} x^{(k+1)} & := & \text{argmin}_{x\in\mathbf{R}^n} & f_0(x,y^{(k)}) \\ & & \text{subject to} & f_i(x,y^{(k)}) \leq 0, \quad i=1,\ldots,m \\ & & & h_i(x,y^{(k)})=0, \quad i=1,\ldots,p, \end{array} \tag{C.2}$$

where the data $y^{(k)} \in \mathbf{R}^k$ is from the previous iteration, and

$$\begin{array}{rcll} y^{(k+1)} & := & \text{argmin}_{y\in\mathbf{R}^k} & f_0(x^{(k+1)},y) \\ & & \text{subject to} & f_i(x^{(k+1)},y) \leq 0, \quad i=1,\ldots,m \\ & & & h_i(x^{(k+1)},y)=0, \quad i=1,\ldots,p, \end{array} \tag{C.3}$$

where the data $x^{(k+1)} \in \mathbf{R}^n$ is the solution of (C.2) in the current iteration.

Directly solving the subproblems in (C.2) and (C.3) may lead to numerical problems or slow convergence in practice. One possible reason for this is that the biconvex objective function f_0 can be very 'flat' in some region along certain directions when one block of variables is fixed (an example is shown in figure 3.3,

in the region where x and y are close to zero), where the convex problem solver for subproblems may have difficulty getting sufficient progress in minimizing the objective along the descent direction.

A common technique to alleviate these issues is to add proximal regularization terms to the objective functions of the subproblems. Specifically, in the $(k+1)$th iteration of the ACS procedure, instead of (C.2) and (C.3), the following updates are performed:

$$\begin{array}{lcll} x^{(k+1)} & := & \operatorname{argmin}_{x\in\mathbf{R}^n} & f_0(x,y^{(k)}) + \lambda\|x-x^{(k)}\|_2^2 \\ & & \text{subject to} & f_i(x,y^{(k)}) \le 0, \quad i=1,\ldots,m \\ & & & h_i(x,y^{(k)}) = 0, \quad i=1,\ldots,p, \end{array} \tag{C.4}$$

and

$$\begin{array}{lcll} y^{(k+1)} & := & \operatorname{argmin}_{y\in\mathbf{R}^k} & f_0(x^{(k+1)},y) + \lambda\|y-y^{(k)}\|_2^2 \\ & & \text{subject to} & f_i(x^{(k+1)},y) \le 0, \quad i=1,\ldots,m \\ & & & h_i(x^{(k+1)},y) = 0, \quad i=1,\ldots,p, \end{array} \tag{C.5}$$

where $\lambda \ge 0$ is the regularization coefficient. The proximal terms $\lambda\|x-x^{(k)}\|_2^2$ and $\lambda\|y-y^{(k)}\|_2^2$ can be interpreted as adding a *trust region penalty* to the respective optimization variables, which penalize large changes of the variables between two consecutive iteration. When $\lambda = 0$, the updates (C.4) and (C.5) reduce to (C.2) and (C.3), respectively. When $\lambda > 0$, the problems in (C.4) and (C.5) become strongly convex, which therefore helps to improve numerical stability when solving the convex subproblems, in the price of slightly slower convergence speeds.

C.1.2 Initialization

Note that the ACS procedure requires a feasible starting point $(x^{(0)},y^{(0)}) \in \mathcal{D}$, since otherwise, one of the subproblems (C.2) or (C.3) may be infeasible right at the first iteration. Formally, the corresponding feasibility problem of a biconvex problem in the form (C.1) can be written as

$$\begin{array}{ll} \text{find} & (x,y) \\ \text{subject to} & f_i(x,y) \le 0, \quad i=1,\ldots,m \\ & h_i(x,y) = 0, \quad i=1,\ldots,p \end{array} \tag{C.6}$$

with variables $x \in \mathbf{R}^n$ and $y \in \mathbf{R}^k$. However, solving the feasibility problem (C.6) directly can be as hard as solving the original biconvex problem (C.1).

The following heuristic via relaxation for finding a solution to the feasibility problem (C.6) is often effective in practice:

$$\begin{array}{ll} \text{minimize} & \mathbf{1}^T s + \|t\|_1 \\ \text{subject to} & s \succeq 0 \\ & f_i(x,y) \le s_i, \quad i=1,\ldots,m \\ & h_i(x,y) = t_i, \quad i=1,\ldots,p, \end{array} \tag{C.7}$$

where $x \in \mathbf{R}^n$, $y \in \mathbf{R}^k$ are the (original) problem variables, and $s \in \mathbf{R}^m$, $t \in \mathbf{R}^p$ are the additional variables to relax the constraints in (C.6), which measures the violation of the constraints. The problem (C.7) is always feasible for any $(x, y) \in \mathbf{R}^n \times \mathbf{R}^k$, since by choosing sufficiently large s and t, all constraints can be satisfied. Then to solve (C.7), we can again use the ACS procedure. The full algorithm is given below.

Algorithm C.1 Alternate convex search initialization.

given a starting point $(x^{(0)}, y^{(0)}) \in \mathbf{R}^n \times \mathbf{R}^k$.

$k := 0$.

repeat

1. $(x^{(k+1)}, s^\star, t^\star) := \underset{\substack{x\in\mathbf{R}^n \\ s\in\mathbf{R}^m,\ t\in\mathbf{R}^p}}{\operatorname{argmin}} \left\{ \mathbf{1}^T s + \|t\|_1 \;\middle|\; \begin{array}{l} s \succeq 0 \\ f_i(x, y^{(k)}) \le s_i,\ i = 1, \ldots, m \\ h_i(x, y^{(k)}) = t_i,\ i = 1, \ldots, p \end{array} \right\}$.

   quit with $(x^{(k+1)}, y^{(k)})$ **if** $\mathbf{1}^T s^\star + \|t^\star\|_1 = 0$.

2. $(y^{(k+1)}, s^\star, t^\star) := \underset{\substack{y\in\mathbf{R}^k \\ s\in\mathbf{R}^m,\ t\in\mathbf{R}^p}}{\operatorname{argmin}} \left\{ \mathbf{1}^T s + \|t\|_1 \;\middle|\; \begin{array}{l} s \succeq 0 \\ f_i(x^{(k+1)}, y) \le s_i,\ i = 1, \ldots, m \\ h_i(x^{(k+1)}, y) = t_i,\ i = 1, \ldots, p \end{array} \right\}$.

   quit with $(x^{(k+1)}, y^{(k+1)})$ **if** $\mathbf{1}^T s^\star + \|t^\star\|_1 = 0$.

3. $k := k + 1$.

until maximum iterations are reached.

Note that to initialize algorithm C.1, we only need to choose a point in $\mathbf{R}^n \times \mathbf{R}^k$. If algorithm C.1 quit with $\mathbf{1}^T s^\star + \|t^\star\|_1 = 0$ in some iteration, then the returned point is a feasible point of the original biconvex problem (C.1), which can then be used as a starting point for the ACS procedure. However, we must note that there is no guarantee that the algorithm C.1 will always find a feasible point for any instance of the biconvex problem (C.1). In practice, as a generic practical method, algorithm C.1 seems to work quite well.

C.1.3 Infeasible start

The relaxation method introduced above, which transforms the biconvex feasibility problem (C.6) into (C.7), can be integrated directly into the original biconvex problem (C.1) as penalty terms, so that the ACS procedure can start directly from an infeasible point. Additionally, by allowing constraints to be violated, it is possible that the ACS procedure finds a region with lower objective values that is otherwise inaccessible when starting from a feasible point. Thus this approach may be desirable even if a feasible initial point is known.

Let $s \in \mathbf{R}^m$ and $t \in \mathbf{R}^p$ be the additional variables to relax the inequality and equality constraints of (C.1), respectively. We consider the following relaxed

biconvex problem:

$$\begin{array}{ll} \text{minimize} & f_0(x,y) + \nu(\mathbf{1}^T s + \|t\|_1) \\ \text{subject to} & s \succeq 0 \\ & f_i(x,y) \le s_i, \quad i = 1,\ldots,m \\ & h_i(x,y) = t_i, \quad i = 1,\ldots,p \end{array} \tag{C.8}$$

with variables $x \in \mathbf{R}^n$, $y \in \mathbf{R}^k$, $s \in \mathbf{R}^m$, and $t \in \mathbf{R}^p$, where $\nu > 0$ is a penalty coefficient. Similar to (C.7), the problem (C.8) is always feasible for any $(x,y) \in \mathbf{R}^n \times \mathbf{R}^k$. Moreover, if the original biconvex problem (C.1) is feasible, then for sufficiently large ν, applying ACS to (C.8) will yield a final point $(x^\star, y^\star, s^\star, t^\star)$, such that $\mathbf{1}^T s^\star + \|t^\star\|_1 = 0$, *i.e.*, $(x^\star, y^\star)$ is feasible and partially optimal for (C.1). (Hence, the penalty term $\nu(\mathbf{1}^T s + \|t\|_1)$ is sometimes called an *exact penalty* of the constraint violation.) This idea leads to the following penalty ACS procedure for solving (C.1).

Algorithm C.2 PENALTY ALTERNATE CONVEX SEARCH.

given a starting point $(x^{(0)}, y^{(0)}) \in \mathbf{R}^n \times \mathbf{R}^k$ and sufficiently large $\nu > 0$.

$k := 0$.

repeat

1. $x^{(k+1)} := \underset{x \in \mathbf{R}^n}{\operatorname{argmin}} \left\{ \begin{array}{l|l} f_0(x, y^{(k)}) & s \in \mathbf{R}^m, \quad t \in \mathbf{R}^p, \quad s \succeq 0 \\ \quad +\nu(\mathbf{1}^T s + \|t\|_1) & f_i(x, y^{(k)}) \le s_i, \ i = 1,\ldots,m \\ & h_i(x, y^{(k)}) = t_i, \ i = 1,\ldots,p \end{array} \right\}.$

2. $y^{(k+1)} := \underset{y \in \mathbf{R}^k}{\operatorname{argmin}} \left\{ \begin{array}{l|l} f_0(x^{(k+1)}, y) & s \in \mathbf{R}^m, \quad t \in \mathbf{R}^p, \quad s \succeq 0 \\ \quad +\nu(\mathbf{1}^T s + \|t\|_1) & f_i(x^{(k+1)}, y) \le s_i, \ i = 1,\ldots,m \\ & h_i(x^{(k+1)}, y) = t_i, \ i = 1,\ldots,p \end{array} \right\}.$

3. $k := k + 1$.

until stopping criteria is satisfied.

Compared to the standard ACS procedure (algorithm 3.1), the penalty ACS procedure in algorithm C.2 can take any point in $\mathbf{R}^n \times \mathbf{R}^k$ as the initial point. The same termination criteria as those presented in remark 3.1 can still be used for algorithm C.2.

Note that, again, there is no guarantee that the final point returned by algorithm C.2 is feasible for the original biconvex problem (C.1) (especially when ν is not large enough). In practice, the value of ν can be selected in an ad hoc manner, *i.e.*, one may try to increase ν and resolve (C.8) if the final point returned by algorithm C.2 is still infeasible for (C.1).

Finally, the proximal regularization terms as in (C.4) and (C.5) can be readily integrated into the subproblems in the algorithm C.2 to improve numerical stability.

C.2 Convex-concave procedure

C.2.1 Domain and differentiability

Consider using the CCP procedure (algorithm 3.2) for approximately solving a difference-of-convex problem in the form

$$\begin{array}{ll} \text{minimize} & f_0(x) - g_0(x) \\ \text{subject to} & f_i(x) - g_i(x) \leq 0, \quad i = 1, \ldots, m \end{array} \tag{C.9}$$

with variable $x \in \mathbf{R}^n$, where the functions $f_i \colon \mathbf{R}^n \to \mathbf{R}$ and $g_i \colon \mathbf{R}^n \to \mathbf{R}$ for $i = 0, \ldots, m$ are convex functions. At the $(k+1)$th iteration, the concave part functions g_i of (C.9) for all $i = 1, \ldots, m$ are first linearized at the current point $x^{(k)}$, *i.e.*, the first-order Taylor expansion

$$\hat{g}_{i,x^{(k)}}(x) := g_i(x^{(k)}) + \nabla g_i(x^{(k)})^T (x - x^{(k)}) \tag{C.10}$$

is formed for all $i = 0, \ldots, m$, at the current point $x^{(k)}$. Then the variable x is updated by solving the following convex approximation of (C.9) at the current point $x^{(k)}$:

$$\begin{array}{lll} x^{(k+1)} & := & \begin{array}{ll} \operatorname{argmin}_{x \in \mathbf{R}^n} & f_0(x) - \hat{g}_{0,x^{(k)}}(x) \\ \text{subject to} & f_i(x) - \hat{g}_{i,x^{(k)}}(x) \leq 0, \quad i = 1, \ldots, m, \end{array} \end{array} \tag{C.11}$$

where $\hat{g}_{i,x^{(k)}}$ is given by (C.10).

Domain of the concave parts

In the standard treatment of CCP (*e.g.*, algorithm 3.2), it is often assumed that the functions g_i for all $i = 1, \ldots, m$ have full domain $\mathbf{R}^n$. However, in practice, it is possible that some of these functions are only defined on a subset of $\mathbf{R}^n$, where in such cases some technical issues may occur. (The functions f_i can also have restricted domains, but this is directly handled by the convex problem solver.)

As a basic example, consider the difference-of-convex problem

$$\begin{array}{ll} \text{minimize} & \sqrt{x} \\ \text{subject to} & x \geq -1 \end{array} \tag{C.12}$$

with variable $x \in \mathbf{R}$. The objective function $\sqrt{x}$ is concave with domain $\mathbf{R}_+$, and the solution is obviously $x^\star = 0$. To express the objective of (C.12) in the form of (C.9), we can let $f_0(x) = 0$ and $g_0(x) = -\sqrt{x}$. Now suppose we start the CCP procedure from some feasible initial point $x^{(0)} \geq -1$, then in the next iteration, we solve the following convex approximation of (C.12):

$$\begin{array}{ll} \text{minimize} & \sqrt{x^{(0)}} + \dfrac{1}{2\sqrt{x^{(0)}}}(x - x^{(0)}) \\ \text{subject to} & x \geq -1, \end{array} \tag{C.13}$$

which is a linear program with variable $x \in \mathbf{R}$. Note that when we linearize g_0 to obtain the convex problem (C.13), the implicit domain constraint $x \geq 0$ in (C.12) is removed. Then directly solving (C.13) gives us the next point $x^{(1)} = -1$. However, at this point, the function $g_0(x) = -\sqrt{x}$ is not defined, and therefore the CCP procedure will fail at the next iteration.

This example suggests that implicit domain constraints of the concave parts g_i in (C.9) must be taken into account explicitly when solving the convex approximations in (C.11), so that $x^{(k+1)}$ is guaranteed to lie in the domain of g_i for all $i = 0, \ldots, m$, and the CCP iterations can therefore proceed. There are several options to achieve this in practice.

The first common approach is to make the domain constraints of g_i explicit in the original problem (C.9), *i.e.*, by rewriting the problem as

$$\begin{array}{ll} \text{minimize} & f_0(x) - g_0(x) \\ \text{subject to} & f_i(x) - g_i(x) \leq 0, \quad i = 1, \ldots, m \\ & x \in \mathbf{dom}\, g_i, \quad i = 0, \ldots, m. \end{array} \tag{C.14}$$

Since the domain of the convex functions g_i are convex sets and are presumably representable via convex inequality and equality constraints (which will remain exactly during all CCP iterations), making domain constraints in (C.9) explicit introduces no additional majorization operations.

An alternative approach to handle the domain issues of the concave part functions g_i is to modify the linearization in (C.10) to include a indicator function of the domain of g_i. Specifically, we can replace the update (C.10) by

$$\hat{g}_{i,x^{(k)}}(x) := g_i(x^{(k)}) + \nabla g_i(x^{(k)})^T (x - x^{(k)}) + I_{\mathbf{dom}\, g_i}(x), \tag{C.15}$$

where $I_{\mathbf{dom}\, g_i}$ is the indicator function of the set $\mathbf{dom}\, g_i$, defined as

$$I_{\mathbf{dom}\, g_i}(x) = \begin{cases} 0, & x \in \mathbf{dom}\, g_i \\ \infty, & \text{otherwise.} \end{cases}$$

Since g_i is a convex function, its domain $\mathbf{dom}\, g_i$ is a convex set, so the indicator function $I_{\mathbf{dom}\, g_i}$ is also convex. Thus, the modified linearization (C.15) remains a convex function. It follows that if we replace the linearization (C.10) by (C.15) in the problem (C.11), the resulting problem is still convex.

Differentiability on boundary

Another issue related to the concave part functions g_i in (C.9) with restricted domains is the differentiability of these functions on the boundary of their domains. In particular, since we need to evaluate the gradients $\nabla g_i(x^{(k)})$ to formulate the linearization (C.10) at each iteration of CCP, if the current point $x^{(k)}$ lies on the boundary of $\mathbf{dom}\, g_i$, then the gradient $\nabla g_i(x^{(k)})$ does not exist.

To see an example, consider again the problem (C.12). If we start the CCP procedure from the initial point $x^{(0)} = 0$, which is feasible, then at the first iteration, we need to compute the gradient of $g_0(x) = -\sqrt{x}$ at $x^{(0)} = 0$. However, since g_0 is not differentiable at $x = 0$, the CCP procedure cannot proceed.

To avoid this issue, we can perform a 'damped' step whenever the current point $x^{(k)}$ lies on the boundary of the domain of any concave part function g_i. Specifically, when $\nabla g_i(x^{(k)})$ does not exist, we replace $x^{(k)}$ by

$$x^{(k)} := (1-\alpha)x^{(k)} + \alpha x^{(k-1)}, \tag{C.16}$$

where $\alpha \in (0,1)$ is a small damping factor, and $x^{(k-1)}$ is the point from the previous iteration. If $x^{(0)}$ is in the interior of $\mathbf{dom}\, g_i$ for all $i = 0,\ldots,m$, then $x^{(k-1)}$ will be in the interior as well for all $k = 1,2,\ldots$, so the gradient $\nabla g_i(x^{(k)})$ is guaranteed to exist for all CCP iterations.

The modified CCP iterations with restricted domain and damped steps (C.16) are summarized in algorithm C.3 below.

Algorithm C.3 CONVEX-CONCAVE PROCEDURE WITH RESTRICTED DOMAIN.

given a feasible point $x^{(0)}$ to (C.14) in the interior of $\mathbf{dom}\, g_i$ for all $i = 1,\ldots,m$.

$k := 0$.

repeat

1. **if** $\nabla g_i(x^{(k)})$ does not exist for some $i = 0,\ldots,m$
   then $x^{(k)} := (1-\alpha)x^{(k)} + \alpha x^{(k-1)}$.
2. $\hat{g}_{i,x^{(k)}}(x) := g_i(x^{(k)}) + \nabla g_i(x^{(k)})^T(x - x^{(k)})$ for all $i = 0,\ldots,m$.
3. $x^{(k+1)} := \operatorname*{argmin}_{x\in\mathbf{R}^n} \left\{ f_0(x) - \hat{g}_{0,x^{(k)}}(x) \;\middle|\; \begin{array}{l} f_i(x) - \hat{g}_{i,x^{(k)}}(x) \le 0,\ i = 1,\ldots,m \\ x \in \mathbf{dom}\, g_i,\ i = 0,\ldots,m \end{array} \right\}$.
4. $k := k + 1$.

until stopping criteria is satisfied.

Note that at the final iteration, the algorithm can converge to some point on the boundary of the domains $\mathbf{dom}\, g_i$, $i = 1,\ldots,m$, but $x^{(k)}$ is guaranteed to be in the interior for all intermediate iterations, which is sufficient to guarantee that the linearization (C.10) exists.

C.2.2 Initialization

If the concave part functions g_i in (C.9) all have full domain $\mathbf{R}^n$, there are several possible approaches to initialize the CCP iterations. In particular, those methods introduced in the previous section for initializing the ACS procedure can be readily adapted. On the other hand, if some of the functions g_i have restricted domains, then we must ensure that the initial point lies in the interior of these restricted domains so that the respective linearizations exist, which will require some additional operations.

Initialization with full domain

We first consider the case where the concave part functions g_i in (C.9) all have full domain $\mathbf{R}^n$, and to directly find a feasible starting point for the CCP iterations. To do so, we consider the following penalty feasibility problem:

$$\begin{array}{ll} \text{minimize} & \mathbf{1}^T s \\ \text{subject to} & s \succeq 0 \\ & f_i(x) - g_i(x) \leq s_i, \quad i = 1, \ldots, m \end{array} \tag{C.17}$$

with variables $x \in \mathbf{R}^n$ and $s \in \mathbf{R}^m$. The problem (C.17) is also a difference-of-convex problem, so we can again use CCP to find a solution. Note that since this penalty feasibility problem can always be made feasible by choosing s sufficiently large, so any initial point $x^{(0)} \in \mathbf{R}^n$ can be used to start the CCP iterations. The full algorithm is given as follows.

Algorithm C.4 Convex-concave procedure initialization.

given a starting point $x^{(0)} \in \mathbf{R}^n$.

$k := 0$.

repeat

1. $\hat{g}_{i,x^{(k)}}(x) := g_i(x^{(k)}) + \nabla g_i(x^{(k)})^T (x - x^{(k)})$ for all $i = 1, \ldots, m$.
2. $(x^{(k+1)}, s^\star) := \underset{x \in \mathbf{R}^n,\ s \in \mathbf{R}^m}{\operatorname{argmin}} \left\{ \mathbf{1}^T s \;\middle|\; \begin{array}{l} s \succeq 0 \\ f_i(x) - \hat{g}_{i,x^{(k)}}(x) \leq s_i,\ i = 1, \ldots, m \end{array} \right\}$.

   quit with $x^{(k+1)}$ **if** $\mathbf{1}^T s^\star = 0$.
3. $k := k + 1$.

until maximum iterations are reached.

If algorithm C.4 quits with $\mathbf{1}^T s^\star = 0$ in some iteration, then the returned point $x^{(k+1)}$ is a feasible point of the original difference-of-convex problem (C.9), which can then be used as a starting point for the CCP procedure (algorithm 3.2).

Infeasible start convex-concave procedure

Another option to initialize the CCP iterations when the concave part functions g_i in (C.9) all have full domain $\mathbf{R}^n$ is to integrate the penalty term $\mathbf{1}^T s$ for constraint violation in (C.17) directly into the original difference-of-convex problem (C.9), as in the problem (C.8) for penalty ACS, so that we do not need a feasible starting point.

Specifically, we consider the following relaxed difference-of-convex problem:

$$\begin{array}{ll} \text{minimize} & f_0(x) - g_0(x) + \nu(\mathbf{1}^T s) \\ \text{subject to} & s \succeq 0 \\ & f_i(x) - g_i(x) \leq s_i, \quad i = 1, \ldots, m, \end{array} \tag{C.18}$$

where the variables are $x \in \mathbf{R}^n$ and $s \in \mathbf{R}^m$, and $\nu > 0$ is the penalty coefficient. Then the following penalty CCP algorithm can be used to find an approximate solution of (C.18).

Algorithm C.5 PENALTY CONVEX-CONCAVE PROCEDURE.

given a starting point $x^{(0)} \in \mathbf{R}^n$ and sufficiently large $\nu > 0$.

$k := 0$.

repeat

1. $\hat{g}_{i,x^{(k)}}(x) := g_i(x^{(k)}) + \nabla g_i(x^{(k)})^T(x - x^{(k)})$ for all $i = 0, \ldots, m$.
2. $x^{(k+1)} := \underset{x \in \mathbf{R}^n}{\operatorname{argmin}} \left\{ \begin{array}{c|l} f_0(x) - \hat{g}_{0,x^{(k)}}(x) & s \in \mathbf{R}^m, \quad s \succeq 0 \\ +\nu(\mathbf{1}^T s) & f_i(x) - \hat{g}_{i,x^{(k)}}(x) \leq s_i, \ i = 1, \ldots, m \end{array} \right\}$.
3. $k := k + 1$.

until stopping criteria is satisfied.

If the original difference-of-convex problem (C.9) is feasible, then for sufficiently large ν, applying the penalty CCP procedure will yield a final point $(x^\star, s^\star)$ such that $\mathbf{1}^T s^\star = 0$, so $x^\star$ is a feasible stationary point for (C.9).

Initialization with restricted domain

Finally we consider the case where some of the concave part functions g_i in (C.9) have restricted domains. Based on the previous discussions, we can combine the basic ideas from algorithm C.3 and algorithm C.5 to obtain a penalty CCP procedure that can start from an infeasible point while taking care of the domain and differentiability issues of the concave parts. The only unsolved issue then is how to ensure that the initial point lies in the interior of the domains of all concave part functions g_i.

While this problem can be very difficult in the most general case, one simple heuristic turns out to work quite well in practice. Suppose we generate random points $z_1, \ldots, z_p \in \mathbf{R}^n$ independently from, *e.g.*, a standard Gaussian distribution, and for each z_j, we project it onto the domain of each concave part function g_i for $i = 0, \ldots, m$, *i.e.*, we solve

$$\begin{array}{ll} \text{minimize} & \|x - z_j\|_2 \\ \text{subject to} & x \in \mathbf{dom}\, g_i, \quad i = 0, \ldots, m, \end{array} \tag{C.19}$$

where the problem variable is $x \in \mathbf{R}^n$. Since g_i are convex functions, their domains $\mathbf{dom}\, g_i$ are convex sets, so the above projection problem is a convex optimization problem. Let $x_j^{(0)}$ be a solution to the projection problem (C.19) for the random point z_j. When $z_j \notin \bigcap_{i=0}^m \mathbf{dom}\, g_i$, the projected point $x_j^{(0)}$ will lie on the boundary of at least one of the domains $\mathbf{dom}\, g_i$, $i = 0, \ldots, m$. Then we take a convex combination of the points $x_1^{(0)}, \ldots, x_p^{(0)}$ to obtain the initial point

$$x^{(0)} = \frac{1}{p} \sum_{j=1}^{p} x_j^{(0)}.$$

Since the domains $\mathbf{dom}\, g_i$ are convex sets, the initial point $x^{(0)}$ obtained in this way is guaranteed to at least satisfy $x^{(0)} \in \mathbf{dom}\, g_i$ for all $i = 0, \ldots, m$, and we might also expect that it is actually in the interior of these domains. Of course, this would not always happen and in this case, such a $x^{(0)}$ will be an unacceptable starting point for the CCP iterations. Nevertheless, as a simple heuristic, this approach seems to work very well.

C.3 Sequential convex approximation

There are numerous technical issues (potentially even more than those might appear in ACS and CCP) that may raise up in actually implementing an SCA algorithm for solving some general nonlinear problem, although each step is, in principle, just solving a convex optimization problem. Some universal challenges in SCA applications include, for example, the convex approximations might be infeasible, how to choose the trust region, or how to evaluate the quality of each step, etc. This section presents some basic ideas that can be potentially used to address these issues.

C.3.1 Penalty methods

As we have seen in the previous sections, penalty methods can be very useful in practice for dealing with feasibility issues in various optimization procedures, and this holds for the SCA procedure as well. Suppose we are attempting to solve the general nonlinear problem

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \ldots, m \\ & h_i(x) = 0, \quad i = 1, \ldots, p, \end{array} \tag{C.20}$$

with variable $x \in \mathbf{R}^n$, where the functions $f_i \colon \mathbf{R}^n \to \mathbf{R}$ for $i = 0, \ldots, m$ is possibly nonconvex and $h_i \colon \mathbf{R}^n \to \mathbf{R}$ for $i = 1, \ldots, p$ is possibly nonaffine. Instead of directly applying the SCA procedure to (C.20) and enforcing the constraints (which might be infeasible anyway), we can consider the following relaxed problem:

$$\begin{array}{ll} \text{minimize} & f_0(x) + \nu(\mathbf{1}^T s + \|t\|_1) \\ \text{subject to} & s \succeq 0 \\ & f_i(x) \leq s_i, \quad i = 1, \ldots, m \\ & h_i(x) = t_i, \quad i = 1, \ldots, p \end{array}$$

where the variables are $x \in \mathbf{R}^n$, $s \in \mathbf{R}^m$, and $t \in \mathbf{R}^p$, and $\nu > 0$ is a penalty coefficient. This problem can be expressed in a more compact form as

$$\text{minimize} \quad \phi(x) = f_0(x) + \nu \left(\sum_{i=1}^m f_i(x)_+ + \sum_{i=1}^p |h_i(x)| \right), \tag{C.21}$$

where $f_i(x)_+ = \max\{f_i(x), 0\}$ are the violations of the inequality constraints. The function $\phi \colon \mathbf{R}^n \to \mathbf{R}$ in (C.21) is often called a *merit function* in the context of

penalty methods for constrained optimization. Then applying the SCA procedure to (C.21) leads to the following convex approximation at the current point $\tilde{x}$:

$$\begin{array}{ll} \text{minimize} & \hat{\phi}_{\tilde{x}}(x) = \hat{f}_{0,\tilde{x}}(x) + \nu\left(\sum_{i=1}^m \hat{f}_{i,\tilde{x}}(x)_+ + \sum_{i=1}^p |\hat{h}_{i,\tilde{x}}(x)|\right) \\ \text{subject to} & x \in \mathcal{T}_{\tilde{x}}, \end{array} \tag{C.22}$$

where $\hat{f}_{i,\tilde{x}}$ and $\hat{h}_{i,\tilde{x}}$ are the convex and affine approximations of f_i and h_i at the current point $\tilde{x}$, respectively, and $\mathcal{T}_{\tilde{x}}$ is the current trust region at $\tilde{x}$. Now the problem (C.22) is always feasible and typically very easy to solve.

In the most general case, solving a nonlinear program (which might even be infeasible) in the form (C.20) via the penalty SCA (C.22) has the following interpretation: With the current point $\tilde{x}$, we are trying to find a new point x in the trust region $\mathcal{T}_{\tilde{x}}$ so that either the objective value $f_0(x)$ is decreased, or the total constraint violation $\sum_{i=1}^m f_i(x)_+ + \sum_{i=1}^p |h_i(x)|$ is reduced, or both. It follows from this interpretation that the penalty SCA method is not guaranteed to decrease the original objective value f_0 at every iteration, since sometimes we might need to increase f_0 as a compromise to reduce the constraint violation. By iteratively applying the above SCA step, we can expect to eventually find a feasible point for the original problem (C.20) if it is feasible, and otherwise, we might still have some information about how much constraint violation is unavoidable.

In other words, for some general nonlinear problems whose feasibility is unknown, we usually resolve our goal of 'attempting to solve the problem' to exploring, to which extend the primary objective can be minimized, and in the mean time how much constraint violation can be avoided. For this purpose, the penalty SCA method provides a practical way allowing us to make some explorations in the direction.

C.3.2 Trust region updates

According to the examples in §3.3.2, the size of the trust region $\mathcal{T}_{\tilde{x}}$ at the current point $\tilde{x}$ plays an important role in the performance of the SCA procedure. Now we discuss some basic ideas for adaptively updating the trust region during the SCA iterations. Let us assume that the trust region at the $(k+1)$th iteration has the form

$$\mathcal{T}^{(k)} = \{x \in \mathbf{R}^n \mid \|x - x^{(k)}\| \leq \rho\},$$

where $\|\cdot\|$ is some norm on $\mathbf{R}^n$ and $\rho > 0$ is the trust region 'radius'.

A common approach to update the trust region radius ρ is based on the *predicted reduction* and *actual reduction* of the merit function ϕ of the convex approximation (C.21) at each SCA iteration. Specifically, at the $(k+1)$th iteration, let $\hat{x}$ be the solution to the convex approximation (C.22) at the current point $x^{(k)}$. Then the predicted reduction is defined as

$$\hat{\delta} = \phi(x^{(k)}) - \hat{\phi}_{x^{(k)}}(\hat{x}),$$

and the actual reduction is defined as

$$\delta = \phi(x^{(k)}) - \phi(\hat{x}).$$

Now we can use the ratio between the actual and predicted reductions to update the trust region radius ρ:

- If $\delta \geq \alpha\hat{\delta}$ for some $\alpha \in (0,1)$, then we increase the trust region radius ρ by some factor $\beta^{\text{succ}} \geq 1$, and accept this $\hat{x}$.
- If $\delta < \alpha\hat{\delta}$ for some $\alpha \in (0,1)$, then we decrease the trust region radius ρ by some factor $\beta^{\text{fail}} \in (0,1)$, and reject this step.

This adaptive trust region update strategy can be interpreted as follows: If the actual decrease is more (or less) than some fraction of the predicted decrease, then we increase (or decrease) the trust region radius for the next iteration.

Algorithm C.6 summarizes these ideas of penalty SCA and adaptive trust region updates.

Algorithm C.6 PENALTY SEQUENTIAL CONVEX APPROXIMATION.

given $x^{(0)} \in \mathbf{R}^n$, $\alpha \in (0,1)$, $\beta^{\text{succ}} \geq 1$, $\beta^{\text{fail}} \in (0,1)$, and $\rho > 0$.

$k := 0$.

repeat

1. *Form (C.22) and solve.* $\hat{x} := \operatorname*{argmin}_{x \in \mathbf{R}^n} \left\{ \hat{\phi}_{x^{(k)}}(x) \;\middle|\; \|x - x^{(k)}\| \leq \rho \right\}$.
2. *Calculate predicted reduction.* $\hat{\delta} := \phi(x^{(k)}) - \hat{\phi}_{x^{(k)}}(\hat{x})$.
3. *Calculate actual reduction.* $\delta := \phi(x^{(k)}) - \phi(\hat{x})$.
4. *Update trust region.*
   if $\delta \geq \alpha\hat{\delta}$
   then $x^{(k+1)} := \hat{x}$, $\rho := \beta^{\text{succ}}\rho$
   else $x^{(k+1)} := x^{(k)}$, $\rho := \beta^{\text{fail}}\rho$.
5. $k := k + 1$.

until stopping criteria is satisfied.

Bibliographical notes

Adding proximal regularization terms to ACS methods for better numerical properties is more or less standard in practice; see, *e.g.*, [BPC+11] and [PB14]. It is also observed that adding the additional proximal regularizers can sometimes lead to better final points [SDU+17], compared to those from the original ACS procedure.

For more comprehensive discussions of the exact penalty methods for constrained optimization problems, *e.g.*, used in the problem (C.8), (C.18), and (C.21), we refer the readers to the book by Nocedal and Wright [NW06, chapters 15 and 17], which also contains good material for many other useful penalty methods, *e.g.*, *quadratic penalty methods*, *(augmented) Lagrangian methods*, for handling constrained optimization problems. The exact penalty approach can be easily adapted when the inequality constraints in involve generalized inequalities, *e.g.*, semidefinite constraints; see [SDU+17] and [ZB25] for more details.

The ideas of integrating the damped step (C.16) into CCP to deal with the differentiability problem at the boundaries of restricted domain were originally proposed by Shen *et al.* [SDGB16]. This paper also introduces the random projection method (*i.e.*, via the problem (C.19)) for generating initial points in the interior of the domains of the concave part functions.

There are many other possible strategies for adaptively updating the trust region in SCA methods; see, *e.g.*, [CGT00] and [NW06, chapter 4] for more details.

Notation

Sets and vector spaces

$\mathbf{R}$, $\mathbf{R}_+$, $\mathbf{R}_{++}$	Reals, nonnegative reals, positive reals.
$\mathbf{Z}$, $\mathbf{Z}_+$, $\mathbf{Z}_{++}$	Integers, nonnegative integers, positive integers.
$\mathbf{R}^n$	Real n-vectors ($n \times 1$ matrices).
$\mathbf{R}^{m\times n}$	Real $m \times n$ matrices.
$\mathbf{S}^n$	Symmetric $n \times n$ matrices.
$\mathbf{S}^n_+$, $\mathbf{S}^n_{++}$	Symmetric positive semidefinite, positive definite, $n \times n$ matrices.
$\mathcal{B}(c, r)$	Euclidean ball with center c and radius r.
$\lvert C\rvert$	Cardinality of set C.
$\mathbf{conv}\, C$	Convex hull of set C.
I_C	Indicator function of set C.
$\mathbf{dist}(C, D)$	Distance between sets (or points) C and D.

Vectors and matrices

$\mathbf{1}$	Vector with all components one.
e_i	ith standard basis vector.
I	Identity matrix.
$x^T y$	Inner product of vectors x and y.
X^T	Transpose of matrix X.
$\mathbf{tr}\, X$	Trace of matrix X.
$\lambda_i(X)$	ith largest eigenvalue of symmetric matrix X.
$\lambda_{\max}(X)$, $\lambda_{\min}(X)$	Maximum, minimum eigenvalue of symmetric matrix X.
$\sigma_i(X)$	ith largest singular value of matrix X.
$\sigma_{\max}(X)$, $\sigma_{\min}(X)$	Maximum, minimum singular value of matrix X.
$x \perp y$	Vectors x and y are orthogonal: $x^T y = 0$.
$x^\perp$	Orthogonal complement of vector x: $\{y \mid x^T y = 0\}$.

$\mathbf{card}\, x$	Cardinality (number of nonzero entries) of vector (or matrix) x.
$\mathbf{diag}(x)$	Diagonal matrix with diagonal entries $x_1, \ldots, x_n$.
$\mathbf{rank}\, A$	Rank of matrix A.
$\mathcal{R}(A)$	Range of matrix A.
$\mathcal{N}(A)$	Nullspace of matrix A.

Generalized inequalities

$x \preceq y$	Componentwise inequality between vectors x and y: all entries of $y - x$ is nonnegative.
$x \prec y$	Strict componentwise inequality between vectors x and y: all entries of $y - x$ is positive.
$X \preceq Y$	Matrix inequality between symmetric matrices X and Y: the matrix $Y - X$ is symmetric positive semidefinite.
$X \prec Y$	Strict matrix inequality between symmetric matrices X and Y: the matrix $Y - X$ is symmetric positive definite.

Norms

$\|\cdot\|$	A norm.
$\|x\|_1$	ℓ_1-norm of vector x.
$\|x\|_2$	Euclidean (or ℓ_2-) norm of vector x.
$\|x\|_\infty$	ℓ_∞-norm of vector x.
$\|x\|_p$	ℓ_p-norm of vector x.
$\|X\|_{\mathrm{sav}}$	Sum of absolute values of entries in matrix X.
$\|X\|_{\mathrm{mav}}$	Maximum absolute value of entries in matrix X.
$\|X\|_2$	Spectral norm (maximum singular value) of matrix X.
$\|X\|_*$	Nuclear norm (sum of singular values) of matrix X.
$\|X\|_F$	Frobenius norm of matrix X.

Functions and functional operators

$f\colon A \to B$	f is a function on the set $\mathbf{dom}\, f \subseteq A$ into the set B.
$\mathbf{dom}\, f$	Domain of function f.
$\mathbf{epi}\, f$	Epigraph of function f.
$\mathbf{hypo}\, f$	Hypograph of function f.
$\mathbf{conv}\, f$	Convex envelope of function f.
∇f	Gradient of function f.
$\nabla^2 f$	Hessian of function f.

$f \circ g$	Composition of functions f and g.
$f \,\square\, g$	Infimal convolution of functions f and g.

Statistics and probability

$\mathbf{E}\, X$	Expected value of random variable X.
$\mathbf{prob}(S)$	Probability of event S.
$\mathbf{var}\, X$	Variance of random variable X.
$\mathcal{N}(\mu, \Sigma)$	Gaussian (normal) distribution with mean μ and covariance (matrix) Σ.
Φ	Cumulative distribution function of a Gaussian $\mathcal{N}(0, 1)$ density.

References

[AB20] A. Agrawal and S. Boyd. Disciplined quasiconvex programming. *Optimization Letters*, 14:1643–1657, 2020.

[Abb15] S. Abbott. *Understanding Analysis*. Springer, 2015.

[ADB19] A. Agrawal, S. Diamond, and S. Boyd. Disciplined geometric programming. *Optimization Letters*, 13:961–976, 2019.

[ADHP09] D. Aloise, A. Deshpande, P. Hansen, and P. Popat. NP-hardness of Euclidean sum-of-squares clustering. *Machine Learning*, 75:245–248, 2009.

[ADLV11] M. Andersen, J. Dahl, Z. Liu, and L. Vandenberghe. Interior-point methods for large-scale cone programming. In S. Sra, S. Nowozin, and S. J. Wright, editors, *Optimization for Machine Learning*, Neural Information Processing Series. MIT Press, 2011.

[ADV04] M. S. Andersen, J. Dahl, and L. Vandenberghe. CVXOPT: A Python package for convex optimization, 2004.

[AGH+19] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl. CasADi — A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1):1–36, 2019.

[AH86] R. J. Aumann and S. Hart. Bi-convexity and bi-martingales. *Israel Journal of Mathematics*, 54(2):159–180, 1986.

[AKDB15] A. Ali, J. Z. Kolter, S. Diamond, and S. Boyd. Disciplined convex stochastic programming: A new framework for stochastic optimization. In *Proceedings of the 31st Conference on Uncertainty in Artificial Intelligence*, pages 62–71. AUAI Press, 2015.

[And71] T. W. Anderson. Estimation of covariance matrices with linear structure and moving average processes of finite order. Technical report no. 6., Stanford University, 1971.

[Apo67] T. M. Apostol. *Calculus, Volume 1*. John Wiley & Sons, 2nd edition, 1967.

[APTJ95] P. Anttila, P. Paatero, U. Tapper, and O. Järvinen. Source identification of bulk wet deposition in Finland by positive matrix factorization. *Atmospheric Environment*, 29(14):1705–1718, 1995.

[AVDB18] A. Agrawal, R. Verschueren, S. Diamond, and S. Boyd. A rewriting system for convex optimization problems. *Journal of Control and Decision*, 5(1):42–60, 2018.

[AW13] T. Achterberg and R. Wunderling. Mixed integer programming: Analyzing 12 years of progress. In M. Jünger and G. Reinelt, editors, *Facets of Combinatorial Optimization: Festschrift for Martin Grötschel*, pages 449–481. Springer, 2013.

[BA96] M. J. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104, 1996.

[Bar19] J. T. Barron. A general and adaptive robust loss function. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4331–4339. IEEE, 2019.

[BB91] S. Boyd and C. Barratt. *Linear Controller Design: Limits of Performance*. Prentice Hall, 1991.

[BBBS11] A. Ben-Tal, S. Bhadra, C. Bhattacharyya, and J. Saketha Nath. Chance constrained uncertain classification via robust optimization. *Mathematical Programming*, 127:145–173, 2011.

[BBC11] D. Bertsimas, D. B. Brown, and C. Caramanis. Theory and applications of robust optimization. *SIAM Review*, 53(3):464–501, 2011.

[BC17] H. H. Bauschke and P. L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. CMS Books in Mathematics. Springer, 2nd edition, 2017.

[BD15] P. J. Bickel and K. A. Doksum. *Mathematical Statistics: Basic Ideas and Selected Topics*. Chapman and Hall/CRC, 2015. Volumes I and II.

[BDH20] E. Burnell, N. B. Damen, and W. Hoburg. GPkit: A human-centered approach to convex optimization in engineering design. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, 2020.

[BEN09] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. *Robust Optimization*. Princeton Series in Applied Mathematics. Princeton University Press, 2009.

[Ber09] D. P. Bertsekas. *Convex Optimization Theory*. Athena Scientific, 2009.

[Ber15] D. P. Bertsekas. *Convex Optimization Algorithms*. Athena Scientific, 2015.

[Ber16] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 3rd edition, 2016.

[BETC22] A. Bambade, S. El-Kazdadi, A. Taylor, and J. Carpentier. PROX-QP: Yet another quadratic programming solver for robotics and beyond. In *RSS 2022 — Robotics: Science and Systems*, 2022.

[BF95] J. V. Burke and M. C. Ferris. A Gauss-Newton method for convex composite optimization. *Mathematical Programming*, 71:179–194, 1995.

[BGSB19] G. Banjac, P. Goulart, B. Stellato, and S. Boyd. Infeasibility detection in the alternating direction method of multipliers for convex optimization. *Journal of Optimization Theory and Applications*, 183(2):490–519, 2019.

[BGT81] R. G. Bland, D. Goldfarb, and M. J. Todd. The ellipsoid method: A survey. *Operations Research*, 29(6):1039–1091, 1981.

[BGV92] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual Workshop on Computational Learning Theory*, pages 144–152. Association for Computing Machinery, 1992.

[BHH+21] M. L. Bynum, G. A. Hackebeil, W. E. Hart, C. D. Laird, B. L. Nicholson, J. D. Siirola, J.-P. Watson, and D. L. Woodruff. *Pyomo — Optimization Modeling in Python*. Springer Optimization and Its Applications. Springer, 2021.

[Bis06] C. M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, 2006.

[Bix12] R. E. Bixby. A brief history of linear and mixed-integer programming computation. In M. Grötschel, editor, *Optimization Stories*, Documenta Mathematica Series, pages 107–121. European Mathematical Society, 2012.

[BJMO12] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Structured sparsity through convex optimization. *Statistical Science*, 27(4):450–468, 2012.

[Bjö96] Å. Björck. *Numerical Methods for Least Squares Problems*. Other Titles in Applied Mathematics. Society for Industrial and Applied Mathematics, 1996.

[BKMR98] A. Brooke, D. Kendrick, A. Meeraus, and R. Raman. *GAMS: A User's Guide*. GAMS Development Corporation, 1998.

[BL06] J. Borwein and A. Lewis. *Convex Analysis and Nonlinear Optimization: Theory and Examples*. CMS Books in Mathematics. Springer, 2nd edition, 2006.

[BL11] J. R. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer Series in Operations Research and Financial Engineering. Springer, 2nd edition, 2011.

[BN98] A. Ben-Tal and A. Nemirovski. Robust convex optimization. *Mathematics of Operations Research*, 23(4):769–805, 1998.

[BN99] A. Ben-Tal and A. Nemirovski. Robust solutions of uncertain linear programs. *Operations Research Letters*, 25(1):1–13, 1999.

[BN01] A. Ben-Tal and A. Nemirovski. *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*. MOS-SIAM Series on Optimization. Society for Industrial and Applied Mathematics, 2001.

[BN02] A. Ben-Tal and A. Nemirovski. Robust optimization — methodology and applications. *Mathematical Programming*, 92:453–480, 2002.

[BNO03] D. P. Bertsekas, A. Nedic, and A. E. Ozdaglar. *Convex Analysis and Optimization*. Athena Scientific, 2003.

[Bor13] K. C. Border. Alternative linear inequalities, 2013. Lecture notes on selected topics in mathematical economics, California Institute of Technology. Version 2020.10.15::09.50.

[Boy11] S. Boyd. Chance constrained optimization, 2011. Lecture slides of EE364a, Stanford University, winter quarter, 2011.

[Boy14] S. Boyd. Subgradient methods, 2014. Lecture notes of EE364b, Stanford University, spring quarter, 2014. With help from J. Park. Based on notes from January 2007.

[BPC+11] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

[BS07] H.-G. Beyer and B. Sendhoff. Robust optimization — A comprehensive survey. *Computer Methods in Applied Mechanics and Engineering*, 196(33-34):3190–3218, 2007.

[BT95] P. T. Boggs and J. W. Tolle. Sequential quadratic programming. *Acta Numerica*, 4:1–51, 1995.

[BT97] D. Bertsimas and J. N. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, 1997.

[Bub15] S. Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

[Bur85] J. V. Burke. Descent methods for composite nondifferentiable optimization problems. *Mathematical Programming*, 33:260–279, 1985.

[BV04] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[BV18] S. Boyd and L. Vandenberghe. *Introduction to Applied Linear Algebra: Vectors, Matrices, and Least Squares*. Cambridge University Press, 2018.

[BXM03] S. Boyd, L. Xiao, and A. Mutapcic. Subgradient methods, 2003. Lecture notes of EE392o, Stanford University, autumn quarter, 2003.

[CA25] G. M. Chari and B. Açıkmeşe. QOCO: A quadratic objective conic optimizer with custom solver generation. *arXiv*, 2503.12658, 2025.

[CB24] G. Casella and R. L. Berger. *Statistical Inference*. Texts in Statistical Science. Chapman and Hall/CRC, 2nd edition, 2024.

[CC59] A. Charnes and W. W. Cooper. Chance-constrained programming. *Management Science*, 6(1):73–79, 1959.

[CCS10] J.-F. Cai, E. J. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

[CDS01] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Review*, 43(1):129–159, 2001.

[CG23] Y. Chen and P. Goulart. An efficient IPM implementation for a class of nonsymmetric cones. *arXiv*, 2305.12275, 2023.

[CGGS98] S. Chandrasekaran, G. H. Golub, M. Gu, and A. H. Sayed. Parameter estimation in the presence of bounded data uncertainties. *SIAM Journal on Matrix Analysis and Applications*, 19(1):235–252, 1998.

[CGT00] A. R. Conn, N. I. M. Gould, and P. L. Toint. *Trust Region Methods*. MOS-SIAM Series on Optimization. Society for Industrial and Applied Mathematics, 2000.

[CGT11a] C. Cartis, N. I. M. Gould, and P. L. Toint. Adaptive cubic regularisation methods for unconstrained optimization. Part I: Motivation, convergence and numerical results. *Mathematical Programming*, 127:245–295, 2011.

[CGT11b] C. Cartis, N. I. M. Gould, and P. L. Toint. Adaptive cubic regularisation methods for unconstrained optimization. Part II: Worst-case function- and derivative-evaluation complexity. *Mathematical Programming*, 130:295–319, 2011.

[Cha04] A. Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, 20:89–97, 2004.

[CLMW11] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM*, 58(3):11:1–11:37, 2011.

[Com09] C. W. Commander. Maximum cut problem, MAX-CUT. In C. A. Floudas and P. M. Pardalos, editors, *Encyclopedia of Optimization*, pages 1991–1999. Springer, 2nd edition, 2009.

[CR09] E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9:717–772, 2009.

[CR12] E. Candès and B. Recht. Exact matrix completion via convex optimization. *Communications of the ACM*, 55(6):111–119, 2012.

[CRPW12] V. Chandrasekaran, B. Recht, P. A. Parrilo, and A. S. Willsky. The convex geometry of linear inverse problems. *Foundations of Computational Mathematics*, 12:805–849, 2012.

[CT10] E. J. Candès and T. Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.

[CT13] A. C. Cameron and P. K. Trivedi. *Regression Analysis of Count Data*. Econometric Society Monographs. Cambridge University Press, 2nd edition, 2013.

[CV95] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995.

[CWA09] S. Coxe, S. G. West, and L. S. Aiken. The analysis of count data: A gentle introduction to Poisson regression and its alternatives. *Journal of Personality Assessment*, 91(2):121–136, 2009.

[DB16] S. Diamond and S. Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

[DCB13] A. Domahidi, E. Chu, and S. Boyd. ECOS: An SOCP solver for embedded systems. In *European Control Conference*, pages 3071–3076. IEEE, 2013.

[De 94] J. De Leeuw. Block-relaxation algorithms in statistics. In *Information Systems and Data Analysis: Prospects—Foundations—Applications*, pages 308–324. Springer, 1994.

[De 05] J. De Leeuw. Applications of convex analysis to multidimensional scaling. In J. R. Barra, F. Brodeau, G. Romier, and B. Van Cutsem, editors, *Recent Developments in Statistics*. North Holland Publishing, 2005. Originally presented at the European Meeting of Statisticians, 6–11 September, 1976, in Grenoble, France.

[Deb59] G. Debreu. *Theory of Value: An Axiomatic Analysis of Economic Equilibrium*. Yale University Press, 1959.

[DFJ54] G. Dantzig, R. Fulkerson, and S. Johnson. Solution of a large-scale traveling-salesman problem. *Journal of the Operations Research Society of America*, 2(4):393–410, 1954.

[DHL17] I. Dunning, J. Huchette, and M. Lubin. JuMP: A modeling language for mathematical optimization. *SIAM Review*, 59(2):295–320, 2017.

[DHS00] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley & Sons, 2nd edition, 2000.

[DJ14] N. Dinh and V. Jeyakumar. Farkas' lemma: three decades of generalizations for mathematical optimization. *Top*, 22:1–22, 2014.

[DL18] D. Drusvyatskiy and A. S. Lewis. Error bounds, quadratic growth, and linear convergence of proximal methods. *Mathematics of Operations Research*, 43(3):919–948, 2018.

[DLR77] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

[Dom13] A. Domahidi. *Methods and Tools for Embedded Optimization and Control*. PhD thesis, ETH Zurich, 2013.

[DTB17] S. Diamond, R. Takapoui, and S. Boyd. A general system for heuristic minimization of convex functions over non-convex sets. *Optimization Methods and Software*, 33(1):165–193, 2017.

[DW78] J. E. Dennis Jr. and R. E. Welsch. Techniques for nonlinear least squares and robust regression. *Communications in Statistics-Simulation and Computation*, 7(4):345–359, 1978.

[EHN96] H. W. Engl, M. Hanke, and A. Neubauer. *Regularization of Inverse Problems*. Springer, 1996.

[EL97] L. El Ghaoui and H. Lebret. Robust solutions to least-squares problems with uncertain data. *SIAM Journal on Matrix Analysis and Applications*, 18(4):1035–1064, 1997.

[EM75] J. Elzinga and T. G. Moore. A central cutting plane algorithm for the convex programming problem. *Mathematical Programming*, 8:134–145, 1975.

[EOL98] L. El Ghaoui, F. Oustry, and H. Lebret. Robust solutions to uncertain semidefinite programs. *SIAM Journal on Optimization*, 9(1):33–52, 1998.

[EY36] C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

[Far02] J. Farkas. Theorie der einfachen Ungleichungen. *Journal für die reine und angewandte Mathematik*, 124:1–27, 1902.

[Faz02] M. Fazel. *Matrix Rank Minimization with Applications*. PhD thesis, Stanford University, 2002.

[Fes58] H. Feshbach. Unified theory of nuclear reactions. *Annals of Physics*, 5(4):357–390, 1958.

[FGK90] R. Fourer, D. M. Gay, and B. W. Kernighan. A modeling language for mathematical programming. *Management Science*, 36(5):519–554, 1990.

[FHB03] M. Fazel, H. Hindi, and S. P. Boyd. Log-det heuristic for matrix rank minimization with applications to Hankel and Euclidean distance matrices. In *Proceedings of 2003 American Control Conference*, volume 3, pages 2156–2162. IEEE, 2003.

[FIC26] FICO. *Xpress Optimization Help*, 2026.

[Fis22] R. A. Fisher. On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 222(594-604):309–368, 1922.

[Fis25] R. A. Fisher. Theory of statistical estimation. *Mathematical Proceedings of the Cambridge Philosophical Society*, 22(5):700–725, 1925.

[Fle82] R. Fletcher. A model algorithm for composite nondifferentiable optimization problems. *Mathematical Programming Study*, 17:67–76, 1982.

[Fle87] R. Fletcher. *Practical Methods of Optimization*. John Wiley & Sons, 2nd edition, 1987.

[FNB20] A. Fu, B. Narasimhan, and S. Boyd. CVXR: An R package for disciplined convex optimization. *Journal of Statistical Software*, 94:1–34, 2020.

[FS69] J. E. Falk and R. M. Soland. An algorithm for separable nonconvex programming problems. *Management Science*, 15(9):550–569, 1969.

[FW80] R. Fletcher and G. A. Watson. First and second order conditions for a class of nondifferentiable optimization problems. *Mathematical Programming*, 18:291–307, 1980.

[Gal89] D. Gale. *The Theory of Linear Economic Models*. University of Chicago Press, 1989. Originally published in 1960 by The Rand Corporation.

[Gau95] C. F. Gauss. *Theory of the Combination of Observations Least Subject to Errors*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, 1995. Translated by G. W. Stewart from the original treatise written in Latin, 1820.

[GB14] M. Grant and S. Boyd. CVX: MATLAB software for disciplined convex programming, version 2.1, 2014.

[GBC16] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

[GBY06] M. Grant, S. Boyd, and Y. Ye. Disciplined convex programming. In *Global Optimization: From Theory to Implementation*, pages 155–210. Springer, 2006.

[GC24] P. J. Goulart and Y. Chen. Clarabel: An interior-point solver for conic programs with quadratic objectives. *arXiv*, 2405.12762, 2024.

[GCG20] M. Garstka, M. Cannon, and P. Goulart. A clique graph based merging strategy for decomposable SDPs. *IFAC-PapersOnLine*, 53(2):7355–7361, 2020. 21th IFAC World Congress.

[GG86] D. Geman and S. Geman. Bayesian image analysis. In E. Bienenstock, F. F. Soulié, and G. Weisbuch, editors, *Disordered Systems and Biological Organization*, pages 301–319. Springer, 1986.

[GI03] D. Goldfarb and G. Iyengar. Robust convex quadratically constrained programs. *Mathematical Programming*, 97:495–515, 2003.

[GLY96] J.-L. Goffin, Z.-Q. Luo, and Y. Ye. Complexity analysis of an interior cutting plane method for convex feasibility problems. *SIAM Journal on Optimization*, 6(3):638–652, 1996.

[GMT14] V. Gabrel, C. Murat, and A. Thiele. Recent advances in robust optimization: An overview. *European Journal of Operational Research*, 235(3):471–483, 2014.

[GMW19] P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, 2019.

[Gom58] R. E. Gomory. Outline of an algorithm for integer solutions to linear programs. *Bulletin of the American Mathematical Society*, 64:275–278, 1958.

[GPK07] J. Gorski, F. Pfeuffer, and K. Klamroth. Biconvex sets and optimization with biconvex functions: A survey and extensions. *Mathematical Methods of Operations Research*, 66(3):373–407, 2007.

[Gra05] M. Grant. *Disciplined Convex Programming*. PhD thesis, Stanford University, 2005.

[Gri81] A. Griewank. The modification of Newton's method for unconstrained optimization by bounding cubic terms. Technical report, DAMTP, University of Cambridge, 1981.

[Grü07] P. D. Grünwald. *The Minimum Description Length Principle*. MIT Press, 2007.

[GTS+94] K. C. Goh, L. Turan, M. G. Safonov, G. P. Papavassilopoulos, and J. H. Ly. Biaffine matrix inequality properties and computational methods. In *Proceedings of 1994 American Control Conference*, volume 1, pages 850–855. IEEE, 1994.

[Gur26] Gurobi Optimization, LLC. *Gurobi Optimizer Reference Manual*, 2026. Version 13.0.

[GV13] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 4th edition, 2013.

[GW95] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, 1995.

[GW12] P. E. Gill and E. Wong. Sequential quadratic programming methods. In L. Lee and S. Leyffer, editors, *Mixed Integer Nonlinear Programming*, volume 154 of *The IMA Volumes in Mathematics and its Applications*, pages 147–224. Springer, 2012.

[Han98] P. C. Hansen. *Rank-Deficient and Discrete Ill-Posed Problems*. Mathematical Modeling and Computation. Society for Industrial and Applied Mathematics, 1998.

[Han10] P. C. Hansen. *Discrete Inverse Problems*. Fundamentals of Algorithms. Society for Industrial and Applied Mathematics, 2010.

[Har59] P. Hartman. On functions representable as a difference of convex functions. *Pacific Journal of Mathematics*, 9(3):707–713, 1959.

[Hay68] E. V. Haynsworth. Determination of the inertia of a partitioned Hermitian matrix. *Linear Algebra and its Applications*, 1(1):73–81, 1968.

[HJ12] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 2nd edition, 2012.

[HK70] A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[HL93a] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms I: Fundamentals*. Grundlehren der Mathematischen Wissenschaften. Springer, 1993.

[HL93b] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms II: Advanced Theory and Bundle Methods*. Grundlehren der Mathematischen Wissenschaften. Springer, 1993.

[HL01] J.-B. Hiriart-Urruty and C. Lemaréchal. *Fundamentals of Convex Analysis*. Springer, 2001.

[HMC19] R. V. Hogg, J. W. McKean, and A. T. Craig. *Introduction to Mathematical Statistics*. Pearson, 8th edition, 2019.

[Hot33] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417–441, 1933.

[Hot36] H. Hotelling. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377, 1936.

[HP95] R. Horst and P. M. Pardalos, editors. *Handbook of Global Optimization*, volume 2 of *Nonconvex Optimization and Its Applications*. Springer, 1995.

[HPT00] R. Horst, P. M. Pardalos, and N. V. Thoai. *Introduction to Global Optimization*. Nonconvex Optimization and Its Applications. Springer, 2nd edition, 2000.

[HPTD91] R. Horst, T. Q. Phong, N. V. Thoai, and J. De Vries. On solving a D.C. programming problem by a sequence of linear programs. *Journal of Global Optimization*, 1:183–203, 1991.

[HR09] P. J. Huber and E. M. Ronchetti. *Robust Statistics*. John Wiley & Sons, 2nd edition, 2009.

[HT96] R. Horst and H. Tuy. *Global Optimization: Deterministic Approaches*. Springer, 3rd edition, 1996.

[HT99] R. Horst and N. V. Thoai. DC programming: Overview. *Journal of Optimization Theory and Applications*, 103(1):1–43, 1999.

[HTF09] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer, 2nd edition, 2009.

[HTZ24] R. V. Hogg, E. A. Tanis, and D. L. Zimmerman. *Probability and Statistical Inference*. Pearson, 10th edition, 2024.

[Hub64] P. J. Huber. Robust estimation of a location parameter. *Annals of Mathematical Statistics*, 35(1), 1964.

[Hub92] P. J. Huber. Robust estimation of a location parameter. In S. Kotz and N. L. Johnson, editors, *Breakthroughs in Statistics: Methodology and Distribution*, pages 492–518. Springer, 1992.

[HWD+17] D. Hallac, C. Wong, S. Diamond, A. Sharang, R. Sosič, S. Boyd, and J. Leskovec. SnapVX: A network-based convex optimization solver. *Journal of Machine Learning Research*, 18:1–5, 2017.

[Jay57] E. T. Jaynes. Information theory and statistical mechanics. *Physical Review*, 106:620–630, 1957.

[Joh85] F. John. Extremum problems with inequalities as subsidiary conditions. In J. Moser, editor, *Fritz John: Collected Papers*, pages 543–560. Birkhäuser, 1985. First published in 1948.

[Jol02] I. T. Jolliffe. *Principal Component Analysis*. Springer Series in Statistics. Springer, 2nd edition, 2002.

[Jor81] C. Jordan. Sur la série de Fourier. *Comptes Rendus Hebdomadaires des Séances de I'Académie des Sciences*, 92:228–230, 1881.

[Kan60] L. V. Kantorovich. Mathematical methods of organizing and planning production. *Management Science*, 6(4):366–422, 1960. Originally written in Russian and published in 1939, Leningrad State University.

[Kar72] R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

[Kel60] J. E. Kelley Jr. The cutting-plane method for solving convex programs. *Journal of the Society for Industrial and Applied Mathematics*, 8(4):703–712, 1960.

[KMO10] R. H. Keshavan, A. Montanari, and S. Oh. Matrix completion from a few entries. *IEEE Transactions on Information Theory*, 56(6):2980–2998, 2010.

[KN25] F. Kılınç-Karzan and A. Nemirovski. *Essential Mathematics for Convex Optimization*. Cambridge University Press, 2025.

[Koe05] R. Koenker. *Quantile Regression*. Econometric Society Monographs. Cambridge University Press, 2005.

[KT51] H. W. Kuhn and A. W. Tucker. Nonlinear programming. In J. Neyman, editor, *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pages 481–492. University of California Press, 1951.

[Kuh76] H. W. Kuhn. Nonlinear programming. A historical view. In R. W. Cottle and C. E. Lemke, editors, *Nonlinear Programming*, volume 9 of *SIAM-AMS Proceedings*, pages 1–26. American Mathematical Society, 1976.

[Lag53] J. L. Lagrange. *Mécanique Analytique*. Mallet-Bachelier, 1853.

[Lan13] K. Lange. *Optimization*. Springer Texts in Statistics. Springer, 2nd edition, 2013.

[LB16] T. Lipp and S. Boyd. Variations and extension of the convex-concave procedure. *Optimization and Engineering*, 17:263–287, 2016.

[Lec89] Y. G. Leclerc. Constructing simple stable descriptions for image partitioning. *International Journal of Computer Vision*, 3:73–102, 1989.

[LH95] C. L. Lawson and R. J. Hanson. *Solving Least Squares Problems*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, 1995.

[LHY00] K. Lange, D. R. Hunter, and I. Yang. Optimization transfer using surrogate objective functions. *Journal of Computational and Graphical Statistics*, 9(1):1–20, 2000.

[Llo82] S. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[LM18] R. J. Larsen and M. L. Marx. *An Introduction to Mathematical Statistics and Its Applications*. Pearson, 6th edition, 2018.

[LMS+10] Z.-Q. Luo, W.-K. Ma, A. M.-C. So, Y. Ye, and S. Zhang. Semidefinite relaxation of quadratic optimization problems. *IEEE Signal Processing Magazine*, 27(3):20–34, 2010.

[Lof04] J. Lofberg. YALMIP: A toolbox for modeling and optimization in MATLAB. In *Proceedings of the IEEE International Symposium on Computed Aided Control Systems Design*, pages 294–289. IEEE, 2004.

[LR19] R. J. A. Little and D. B. Rubin. *Statistical Analysis with Missing Data*. John Wiley & Sons, 3rd edition, 2019.

[LS71] W. H. Lawton and E. A. Sylvestre. Self modeling curve resolution. *Technometrics*, 13(3):617–633, 1971.

[LS99] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.

[LS00] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In T. Leen, T. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems*, volume 13. MIT Press, 2000.

[LS13] M. Locatelli and F. Schoen. *Global Optimization: Theory, Algorithms, and Applications*. MOS-SIAM Series on Optimization. Society for Industrial and Applied Mathematics, 2013.

[Lue69] D. G. Luenberger. *Optimization by Vector Space Methods*. Series in Decision and Control. John Wiley & Sons, 1969.

[LY08] D. G. Luenberger and Y. Ye. *Linear and Nonlinear Programming*. International Series in Operations Research & Management Science. Springer, 3rd edition, 2008.

[LZ12] S. Lambert-Lacroix and L. Zwald. The BerHu penalty and the grouped effect. *arXiv*, 1207.6868, 2012.

[Mac67] J. B. MacQueen. Some methods of classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pages 281–297. University of California Press, 1967.

[Mac03] D. J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

[Mak20] A. Makhorin. *GNU Linear Programming Kit*, 2020. Version 5.0.

[Man65] O. L. Mangasarian. Linear and nonlinear separation of patterns by linear programming. *Operations Research*, 13(3):444–452, 1965.

[Man94] O. L. Mangasarian. *Nonlinear Programming*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, 1994. Originally published in 1969 by McGraw-Hill.

[Mey23] C. D. Meyer. *Matrix Analysis and Applied Linear Algebra*. Other Titles in Applied Mathematics. Society for Industrial and Applied Mathematics, 2nd edition, 2023.

[MF97] C. D. Maranas and C. A. Floudas. Global optimization in generalized geometric programming. *Computers & Chemical Engineering*, 21(4):351–369, 1997.

[MHT10] R. Mazumder, T. Hastie, and R. Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *Journal of Machine Learning Research*, 11(80):2287–2322, 2010.

[Mir60] L. Mirsky. Symmetric gauge functions and unitarily invariant norms. *The Quarterly Journal of Mathematics*, 11(1):50–59, 1960.

[MK07] G. J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. John Wiley & Sons, 2007.

[MOS26] MOSEK ApS. *MOSEK Optimization Suite*, 2026. Version 11.1.3.

[MT11] J. E. Marsden and A. Tromba. *Vector Calculus*. Worth Publishing Ltd, 2011.

[Mur12] K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT press, 2012.

[Mur22] K. P. Murphy. *Probabilistic Machine Learning: An Introduction*. MIT Press, 2022.

[Mur23] K. P. Murphy. *Probabilistic Machine Learning: Advanced Topics*. MIT Press, 2023.

[Mus24] J. Muscat. *Functional analysis: An Introduction to Metric Spaces, Hilbert Spaces, and Banach Algebras*. Springer, 2024.

[MW65] B. L. Miller and H. M. Wagner. Chance constrained programming with joint constraints. *Operations Research*, 13(6):930–945, 1965.

[MWG95] A. Mas-Colell, M. D. Whinston, and J. R. Green. *Microeconomic Theory*. Oxford University Press, 1995.

[Nem24] A. Nemirovski. *Introduction to Linear Optimization*. World Scientific, 2024.

[Nes98] Y. Nesterov. Semidefinite relaxation and nonconvex quadratic optimization. *Optimization Methods and Software*, 9(1–3):141–160, 1998.

[Nes04] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Springer, 2004.

[NN94] Y. Nesterov and A. Nemirovskii. *Interior-Point Polynomial Algorithms in Convex Programming*. Studies in Applied and Numerical Mathematics. Society for Industrial and Applied Mathematics, 1994.

[NP06] Y. Nesterov and B. T. Polyak. Cubic regularization of Newton method and its global performance. *Mathematical Programming*, 108:177–205, 2006.

[NS06] A. Nemirovski and A. Shapiro. Convex approximations of chance constrained programs. *SIAM Journal on Optimization*, 17(4):969–996, 2006.

[NT08] A. S. Nemirovski and M. J. Todd. Interior-point methods for optimization. *Acta Numerica*, 17:191–234, 2008.

[NW06] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 2nd edition, 2006.

[NWY00] Y. Nesterov, H. Wolkowicz, and Y. Ye. Semidefinite programming relaxations of nonconvex quadratic optimization. In H. Wolkowicz, R. Saigal, and L. Vandenberghe, editors, *Handbook of Semidefinite Programming: Theory, Algorithms, and Applications*, chapter 13, pages 361–419. Springer, 2000.

[OCPB16] B. O'Donoghue, E. Chu, N. Parikh, and S. Boyd. Conic optimization via operator splitting and homogeneous self-dual embedding. *Journal of Optimization Theory and Applications*, 169(3):1042–1068, 2016.

[OCPB23] B. O'Donoghue, E. Chu, N. Parikh, and S. Boyd. SCS: Splitting conic solver, 2023. Version 3.2.11.

[O'D21] B. O'Donoghue. Operator splitting for a homogeneous embedding of the linear complementarity problem. *SIAM Journal on Optimization*, 31:1999–2023, 2021.

[Pap81] C. H. Papadimitriou. On the complexity of integer programming. *Journal of the ACM*, 28(4):765–768, 1981.

[Par14] V. Pareto. *Manual of Political Economy: A Critical and Variorum Edition*. Oxford University Press, 2014. Edited by A. Montesano, A. Zanni, L. Bruni, J. S. Chipman and M. McLure. Originally written in Italian and published in 1906.

[PB14] N. Parikh and S. Boyd. Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3):127–239, 2014.

[Pea94] K. Pearson. Contributions to the mathematical theory of evolution. *Philosophical Transactions of the Royal Society of London. A*, 185:71–110, 1894.

[Pea01] K. Pearson. LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[Phi62] D. L. Phillips. A technique for the numerical solution of certain integral equations of the first kind. *Journal of the ACM*, 9(1):84–97, 1962.

[Pol87] B. T. Polyak. *Introduction to Optimization*. Translations Series in Mathematics and Engineering. Optimization Software, 1987.

[Pow73] M. J. D. Powell. On search directions for minimization algorithms. *Mathematical Programming*, 4(1):193–201, 1973.

[PR02] P. M. Pardalos and H. E. Romeijn, editors. *Handbook of Global Optimization: Volume 2*, volume 62 of *Nonconvex Optimization and Its Applications*. Springer, 2002.

[Pré71] A. Prékopa. Logarithmic concave measures with applications to stochastic programming. *Acta Scientiarum Mathematicarum*, 32:301–316, 1971.

[Pré73] A. Prékopa. On logarithmic concave measures and functions. *Acta Scientiarum Mathematicarum*, 34:335–343, 1973.

[Pré80] A. Prékopa. Logarithmic concave measures and related topics. In M. A. H. Dempster, editor, *Stochastic Programming*, pages 63–82. Academic Press, 1980.

[Pré95] A. Prékopa. *Stochastic Programming*. Kluwer Academic Publishers, 1995.

[PT94] P. Paatero and U. Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126, 1994.

[PTAK91] P. Paatero, U. Tapper, P. Aalto, and M. Kulmala. Matrix factorization methods for analysing diffusion battery data. *Journal of Aerosol Science*, 22(Supplement 1):S273–S276, 1991. Part of *Proceedings of the 1991 European Aerosol Conference*, edited by H. Fissan, W. Höllander, and W. Schütz.

[PW00] F. A. Potra and S. J. Wright. Interior-point methods. *Journal of Computational and Applied Mathematics*, 124(1-2):281–302, 2000.

[Que46] A. Quetelet. *Lettres à S.A.R. le duc régnant de Saxe-Coburg et Gotha: Sur la théorie des probabilités, appliquée aux sciences morales et politiques*. M. Hayez, 1846.

[RFP10] B. Recht, M. Fazel, and P. A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, 52(3):471–501, 2010.

[Rob72] S. M. Robinson. A quadratically-convergent algorithm for general nonlinear programming problems. *Mathematical Programming*, 3:145–156, 1972.

[Roc70] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

[ROF92] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1-4):259–268, 1992.

[Ros65] J. B. Rosen. Pattern separation by convex programming. *Journal of Mathematical Analysis and Applications*, 10:123–134, 1965.

[RU02] R. T. Rockafellar and S. Uryasev. Conditional value-at-risk for general loss distributions. *Journal of Banking & Finance*, 26(7):1443–1471, 2002.

[Rud76] W. Rudin. *Principles of Mathematical Analysis*. McGraw-Hill, 1976.

[RY22] E. K. Ryu and W. Yin. *Large-Scale Convex Optimization: Algorithms & Analyses via Monotone Operators*. Cambridge University Press, 2022.

[SAB23] X. Shen, A. Ali, and S. Boyd. Minimizing oracle-structured composite functions. *Optimization and Engineering*, 24:743–777, 2023.

[Sam59] A. L. Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3(3):210–229, 1959.

[SBG+20] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd. OSQP: An operator splitting solver for quadratic programs. *Mathematical Programming Computation*, 12(4):637–672, 2020.

[SBL20] M. Schubiger, G. Banjac, and J. Lygeros. GPU acceleration of ADMM for large-scale quadratic programming. *Journal of Parallel and Distributed Computing*, 144:55–67, 2020.

[Sch07] E. Schmidt. Zur Theorie der linearen und nichtlinearen Integralgleichungen. *Mathematische Annalen*, 63:433–476, 1907.

[Sch17] J. Schur. Über Potenzreihen, die im Innern des Einheitskreises beschränkt sind. *Journal für die reine und angewandte Mathematik*, 1917(147):205–232, 1917.

[Sch98] A. Schrijver. *Theory of Linear and Integer Programming*. Wiley Series in Discrete Mathematics & Optimization. John Wiley & Sons, 1998.

[SDGB16] X. Shen, S. Diamond, Y. Gu, and S. Boyd. Disciplined convex-concave programming. In *55th IEEE Conference on Decision and Control*, pages 1009–1014. IEEE, 2016.

[SDR21] A. Shapiro, D. Dentcheva, and A. Ruszczyński. *Lectures on Stochastic Programming: Modeling and Theory*. MOS-SIAM Series on Optimization. Society for Industrial and Applied Mathematics, 3rd edition, 2021.

[SDU+17] X. Shen, S. Diamond, M. Udell, Y. Gu, and S. Boyd. Disciplined multi-convex programming. In *29th Chinese Control and Decision Conference*, pages 895–900. IEEE, 2017.

[Ser15] S. A. Serrano. *Algorithms for Unsymmetric Cone Optimization and an Implementation for Problems with the Exponential Cone*. PhD thesis, Stanford University, 2015.

[Sha48] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948.

[Sho85] N. Z. Shor. *Minimization methods for non-differentiable functions*, volume 3 of *Springer Series in Computational Mathematics*. Springer, 1985.

[Sho98] N. Z. Shor. *Nondifferentiable Optimization and Polynomial Problems*, volume 24 of *Nonconvex Optimization and Its Applications*. Springer, 1998.

[SL09] B. K. Sriperumbudur and G. R. G. Lanckriet. On the convergence of the concave-convex procedure. In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc., 2009.

[SLB24] P. Schiele, E. Luxenberg, and S. Boyd. Disciplined saddle programming. *Transactions on Machine Learning Research*, 2024.

[SNB+18] B. Stellato, V. V. Naik, A. Bemporad, P. Goulart, and S. Boyd. Embedded mixed-integer quadratic optimization using the OSQP solver. In *European Control Conference*. IEEE, 2018.

[SS86] F. Santosa and W. W. Symes. Linear inversion of band-limited reflection seismograms. *SIAM Journal on Scientific and Statistical Computing*, 7(4):1307–1330, 1986.

[SS01] B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2001.

[SS22] G. Sagnol and M. Stahlberg. PICOS: A Python interface to conic optimization solvers. *Journal of Open Source Software*, 7(70):3915, 2022.

[Ste56] H. Steinhaus. Sur la division des corps matériels en parties. *Bulletin de l'Académie Polonaise des Sciences, Classe III*, 4(12):801–804, 1956.

[Sti84] S. M. Stigler. Studies in the history of probability and statistics XL Boscovich, Simpson and a 1760 manuscript note on fitting a linear relation. *Biometrika*, 71(3):615–620, 1984.

[Str06] G. Strang. *Linear Algebra and its Applications*. Cengage Learning, 4th edition, 2006.

[SV99] J. A. K. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural Processing Letters*, 9:293–300, 1999.

[SVH05] A. J. Smola, S. V. N. Vishwanathan, and T. Hofmann. Kernel methods for missing variables. In *International Workshop on Artificial Intelligence and Statistics*, pages 325–332. PMLR, 2005.

[Syl51] J. J. Sylvester. XXXVII. On the relation between the minor determinants of linearly equivalent quadratic functions. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 1(4):295–305, 1851.

[TA77] A. N. Tikhonov and V. Y. Arsenin. *Solutions of Ill-Posed Problems*. John Wiley & Sons, 1977. Translated from Russian.

[Tao22] T. Tao. *Analysis I*. Texts and Readings in Mathematics. Springer, 4th edition, 2022.

[TH88] H. Tuy and R. Horst. Convergence and restart in branch-and-bound algorithms for global optimization. Application to concave minimization and D.C. optimization problems. *Mathematical Programming*, 41:161–183, 1988.

[Tib96] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288, 1996.

[TÖ95] O. Toker and H. Özbay. On the NP-hardness of solving bilinear matrix inequalities and simultaneous stabilization with static output feedback. In *Proceedings of 1995 American Control Conference*, volume 4, pages 2525–2526. IEEE, 1995.

[Tuy86] H. Tuy. A general deterministic approach to global optimization via D.C. programming. In *North-Holland Mathematics Studies*, volume 129, pages 273–303. Elsevier, 1986.

[TŽ89] A. Törn and A. Žilinskas, editors. *Global Optimization*, volume 350 of *Lecture Notes in Computer Science*. Springer, 1989.

[UHZB16] M. Udell, C. Horn, R. Zadeh, and S. Boyd. Generalized low rank models. *Foundations and Trends® in Machine Learning*, 9(1):1–118, 2016.

[UMZ+14] M. Udell, K. Mohan, D. Zeng, J. Hong, S. Diamond, and S. Boyd. Convex optimization in Julia. In *Proceedings of the Workshop for High Performance Technical Computing in Dynamic Languages*, pages 18–28, 2014.

[UP01] S. Uryasev and P. M. Pardalos, editors. *Stochastic Optimization: Algorithms and Applications*. Kluwer Academic Publishers, 2001.

[Van10] L. Vandenberghe. The CVXOPT linear and quadratic cone program solvers, 2010.

[Vap98] V. N. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, 1998.

[Vap00] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Information Science and Statistics. Springer, 2nd edition, 2000.

[Vap06] V. Vapnik. *Estimation of Dependences Based on Empirical Data*. Information Science and Statistics. Springer, 2006. Originally published in Russian in 1982. Translated by S. Kotz. Reprinted with afterword.

[VB96] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38(1):49–95, 1996.

[VC64] V. N. Vapnik and A. Ya. Chervonenkis. A note on one class of perceptrons. *Automation and Remote Control*, 25(6):937–945, 1964. Published in Russian.

[War63] J. Warga. Minimizing certain convex functions. *Journal of the Society for Industrial and Applied Mathematics*, 11(3):588–593, 1963.

[WDE07] M. Weiser, P. Deuflhard, and B. Erdmann. Affine conjugate adaptive Newton methods for nonlinear elastomechanics. *Optimisation Methods and Software*, 22(3):413–431, 2007.

[Wei09] T. Weise. *Global Optimization Algorithms: Theory and Application*. Self published, 2009.

[WGR+09] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc., 2009.

[WH76] R. E. Wendell and A. P. Hurter Jr. Minimization of a non-separable objective function subject to disjoint constraints. *Operations Research*, 24(4):643–657, 1976.

[Whi71] P. Whittle. *Optimization under Constraints: Theory and Applications of Nonlinear Programming*. John Wiley & Sons, 1971.

[Wil63] R. B. Wilson. *A Simplicial Algorithm for Concave Programming*. PhD thesis, Graduate School of Business Administration, Harvard University, 1963.

[Wol08] L. A. Wolsey. Mixed integer programming. In B. W. Wah, editor, *Wiley Encyclopedia of Computer Science and Engineering*, pages 1–10. John Wiley & Sons, 2008.

[Wol21] L. A. Wolsey. *Integer Programming*. John Wiley & Sons, 2nd edition, 2021.

[WZ99] H. Wolkowicz and Q. Zhao. Semidefinite programming relaxations for the graph partitioning problem. *Discrete Applied Mathematics*, 96–97:461–479, 1999.

[WZ05] S. W. Wallace and W. T. Ziemba. *Applications of Stochastic Programming*. MOS-SIAM Series on Optimization. Society for Industrial and Applied Mathematics, 2005.

[XCS10] H. Xu, C. Caramanis, and S. Sanghavi. Robust PCA via outlier pursuit. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010.

[XCS12] H. Xu, C. Caramanis, and S. Sanghavi. Robust PCA via outlier pursuit. *IEEE Transactions on Information Theory*, 58(5):3047–3064, 2012.

[YL06] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 68(1):49–67, 2006.

[Yos12] K. Yosida. *Functional Analysis*. Springer, 6th edition, 2012.

[YR03] A. L. Yuille and A. Rangarajan. The concave-convex procedure. *Neural Computation*, 15(4):915–936, 2003.

[ZB25] H. Zhu and J. Boedecker. Disciplined biconvex programming. *arXiv*, 2511.01813, 2025.

[Zha00] S. Zhang. Quadratic maximization and semidefinite relaxation. *Mathematical Programming*, 87:453–465, 2000.

[Zha05] F. Zhang, editor. *The Schur Complement and Its Applications*. Numerical Methods and Algorithms. Springer, 2005.

[Zho21] Z.-H. Zhou. *Machine Learning*. Springer, 2021. Translated by S. Liu from the original book in Chinese published by Tsinghua University Press, 2016.

[ZHZB26] H. Zhu, J. Hoffmann, B. Zhang, and J. Boedecker. Fitting reinforcement learning model to behavioral data under bandits. *Frontiers in Applied Mathematics and Statistics*, 12:1762084, 2026.

[ZOB20] J. Zhang, B. O'Donoghue, and S. Boyd. Globally convergent type–I Anderson acceleration for non-smooth fixed-point iterations. *SIAM Journal on Optimization*, 30(4):3170–3197, 2020.

[ZYHB25] H. Zhu, S. Yan, J. Hoffmann, and J. Boedecker. Multi-convex programming for discrete latent factor models prototyping. *arXiv*, 2504.01431, 2025.

Index