Disciplined Biconvex Programming

Hao Zhu and Joschka Boedecker

Department of Computer Science
University of Freiburg

December 2, 2025

Biconvex programming

$$
\begin{array}{ll}
\text{minimize} & f_0(x,y) \\
\text{subject to} & f_i(x,y) \leq 0, \quad i=1,\ldots,m \\
 & (Ax+b)^T(Cy+d)=0
\end{array}
$$

variables $x \in \mathbf{R}^n$, $y \in \mathbf{R}^k$; $f_0, f_1, \ldots, f_m$ biconvex; $A \in \mathbf{R}^{p\times n}$, $C \in \mathbf{R}^{p\times k}$

applications: approximation, fitting, statistical estimation, *etc.*

- matrix factorization, dictionary learning, latent factor models
- control with bilinear matrix inequalities
- blind deconvolution
- bilinear regression
- . . .

Overview

disciplined biconvex programming (DBCP):

- domain specific language for biconvex programming
- an extension of CVXPY
- solution methods (roughly) based on alternate minimization
- supports fast modeling and prototyping of biconvex problems

1. Biconvex programming

- biconvex sets
- biconvex functions
- operations that preserves biconvexity
- biconvex optimization problems
- partial optimality

Biconvex sets

let $\mathcal{X} \subseteq \mathbf{R}^n$, $\mathcal{Y} \subseteq \mathbf{R}^k$ be nonempty convex sets

$B \subseteq \mathcal{X} \times \mathcal{Y} \subseteq \mathbf{R}^{n+k}$ is a **biconvex set**, if

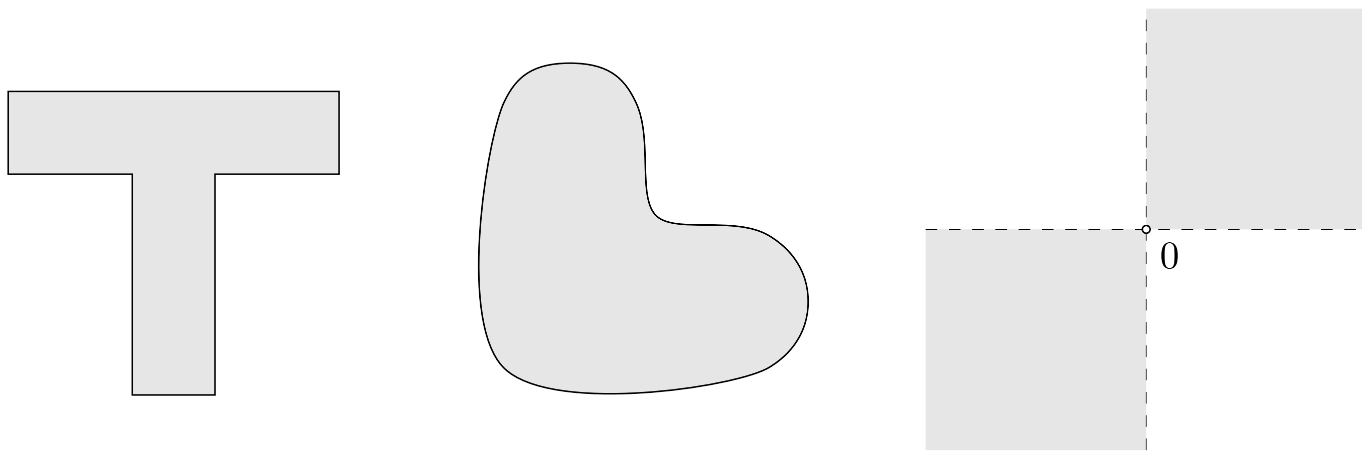
- fix $\tilde{y} \in \mathcal{Y} \implies B_{\tilde{y}} = \{x \in \mathcal{X} \mid (x, \tilde{y}) \in B\} \subseteq \mathbf{R}^n$ convex

- fix $\tilde{x} \in \mathcal{X} \implies B_{\tilde{x}} = \{y \in \mathcal{Y} \mid (\tilde{x}, y) \in B\} \subseteq \mathbf{R}^k$ convex

algebraic property:

- the intersection of (any number of) biconvex sets is biconvex

examples:



- biconvex sets can be unconnected, *e.g.*,

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

Biconvex functions

$f: \mathcal{X} \times \mathcal{Y} \to \mathbf{R}$ is a **biconvex function** if

- $\mathbf{dom}\, f = \{(x, y) \in \mathcal{X} \times \mathcal{Y} \mid f(x, y) < \infty\}$ biconvex
- fix $\tilde{y} \in \mathcal{Y} \implies f_{\tilde{y}}: \mathcal{X} \to \mathbf{R}$, $x \mapsto f(x, \tilde{y})$ convex
- fix $\tilde{x} \in \mathcal{X} \implies f_{\tilde{x}}: \mathcal{Y} \to \mathbf{R}$, $y \mapsto f(\tilde{x}, y)$ convex

biconcave, **biaffine**, and **bilinear** functions are defined similarly

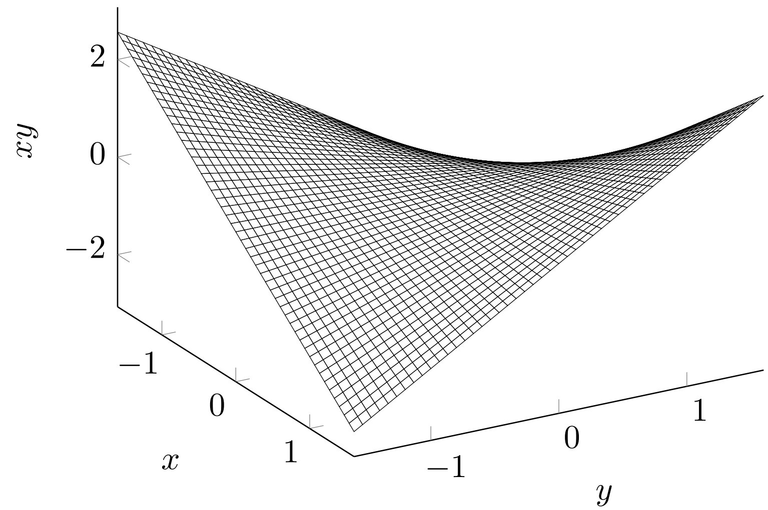
α-**sublevel set** of a biconvex function $f: \mathcal{X} \times \mathcal{Y} \to \mathbf{R}$:

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

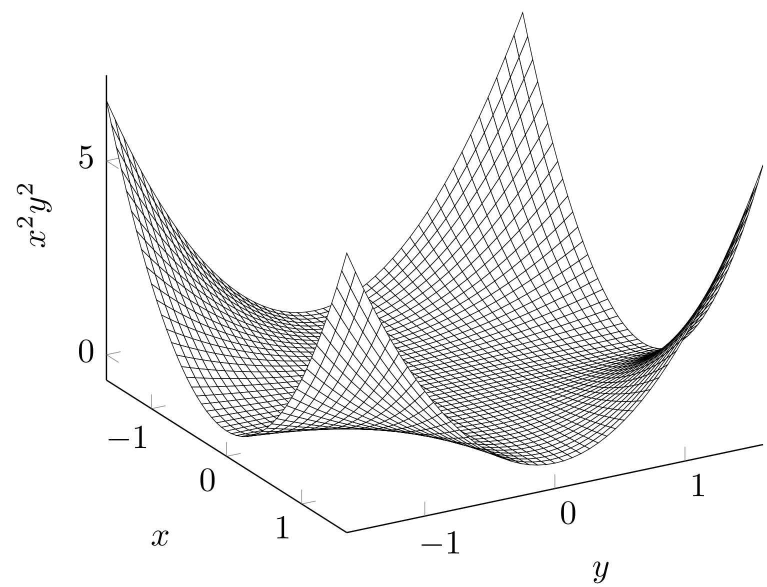
is a biconvex set

examples:

$$f(x, y) = xy$$



$$f(x, y) = x^2y^2$$



Operations that preserve biconvexity

- **nonnegative weighted sum**:
  - $f_1, \ldots, f_m$ biconvex, $w \in \mathbf{R}^m_+ \implies w_1 f_1 + \cdots + w_m f_m$ biconvex
- **pointwise maximum**:
  - $f_1, \ldots, f_m$ biconvex $\implies \max\{f_1, \ldots, f_m\}$ biconvex
  - $\{f_i\}_{i \in \mathcal{I}}$ biconvex ($\mathcal{I}$ an index set) $\implies \sup_{i \in \mathcal{I}} f_i$ biconvex
- **biaffine precomposition**:
  - $h \colon \mathbf{R} \to \mathbf{R}$ convex $\implies h((Ax+b)^T(Cy+d))$ biconvex
    ($A \in \mathbf{R}^{m \times n}$, $C \in \mathbf{R}^{m \times k}$, $b, d \in \mathbf{R}^m$)
- **composition** of $h \colon \mathbf{R} \to \mathbf{R}$ and $g \colon \mathcal{X} \times \mathcal{Y} \to \mathbf{R}$:
  - $h \circ g$ biconvex if
    h convex nondecreasing, g biconvex
    h convex nonincreasing, g biconcave

Biconvex optimization problems

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

- $x \in \mathcal{X}$, $y \in \mathcal{Y}$ are the problem variables
- $f_0, f_1, \dots, f_m \colon \mathcal{X} \times \mathcal{Y} \to \mathbf{R}$ are biconvex
- $h_1, \dots, h_p \colon \mathcal{X} \times \mathcal{Y} \to \mathbf{R}$ are biaffine
- difficult to solve in general (mostly NP-hard)

Partial optimality

let $\mathcal{D}$ be the feasible set of a biconvex problem

$(x^\star, y^\star) \in \mathcal{D}$ is **partially optimal** to a biconvex optimization problem if

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

for all $x \in \mathcal{D}_{y^\star}$, $y \in \mathcal{D}_{x^\star}$

- every stationary point of a differentiable biconvex optimization problem is partially optimal, and vice versa
- partially optimal points are not necessarily globally or even locally optimal
- turn out to work quite well in practical applications

2. Solving biconvex problems

- alternate convex search
- proximal regularization
- initialization
- infeasible start

Alternate convex search (ACS)

basic idea: alternate between convex subproblems in x and y

Algorithm 1 ALTERNATE CONVEX SEARCH.

given a starting point $(x^{(0)}, y^{(0)}) \in \mathcal{D}$

$k := 0$.

repeat

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

2. $y^{(k+1)} := \underset{y \in \mathcal{Y}}{\operatorname{argmin}} \left\{ f_0(x^{(k+1)}, y) \;\middle|\; \begin{array}{l} f_i(x^{(k+1)}, y) \leq 0, \; i = 1, \ldots, m \\ h_i(x^{(k+1)}, y) = 0, \; i = 1, \ldots, p \end{array} \right\}$

3. $k := k + 1$.

until stopping criteria is satisfied.

Proximal regularization

add proximal terms to the ACS subproblems: at the kth iteration, solve

$$\begin{array}{rcll} x^{(k+1)} & := & \mathrm{argmin}_{x\in\mathcal{X}} & f_0(x,y^{(k)}) + \lambda\|x-x^{(k)}\|_2^2 \\ & & \text{subject to} & f_i(x,y^{(k)}) \le 0, \quad i=1,\ldots,m \\ & & & h_i(x,y^{(k)}) = 0, \quad i=1,\ldots,p \end{array}$$

and

$$\begin{array}{rcll} y^{(k+1)} & := & \mathrm{argmin}_{y\in\mathcal{Y}} & f_0(x^{(k+1)},y) + \lambda\|y-y^{(k)}\|_2^2 \\ & & \text{subject to} & f_i(x^{(k+1)},y) \le 0, \quad i=1,\ldots,m \\ & & & h_i(x^{(k+1)},y) = 0, \quad i=1,\ldots,p \end{array}$$

- $\lambda \ge 0$ is the regularization parameter
- large enough $\lambda \implies$ strongly convex subproblems
- better numerical performance and convergence properties

Initialization

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

- as hard as solving the original problem

heuristic via relaxation:

$$
\begin{array}{ll}
\text{minimize} & \mathbf{1}^T s + \|t\|_1 \\
\text{subject to} & s \succeq 0 \\
& f_i(x, y) \leq s_i, \quad i = 1, \ldots, m \\
& h_i(x, y) = t_i, \quad i = 1, \ldots, p
\end{array}
$$

- variables $x \in \mathcal{X}$, $y \in \mathcal{Y}$, $s \in \mathbf{R}^m$, $t \in \mathbf{R}^p$
- if optimal value is zero, then we have found a feasible point
- again, use ACS to solve the relaxation

Infeasible start

$$
\begin{array}{ll}
\text{minimize} & f_0(x,y) + \nu(\mathbf{1}^T s + \|t\|_1) \\
\text{subject to} & s \succeq 0 \\
& f_i(x,y) \leq s_i, \quad i = 1, \ldots, m \\
& h_i(x,y) = t_i, \quad i = 1, \ldots, p
\end{array}
$$

- variables $x \in \mathcal{X}$, $y \in \mathcal{Y}$, $s \in \mathbf{R}^m$, $t \in \mathbf{R}^p$
- $\nu > 0$ is a penalty parameter
- arbitrary starting point $(x^{(0)}, y^{(0)}) \in \mathcal{X} \times \mathcal{Y}$
- large engough ν leads to final points feasible to the original problem

3. Examples

- k-means clustering
- bilinear logistic regression
- sparse dictionary learning
- input-output hidden Markov model

k-means clustering

- given data points $x_i \in \mathbf{R}^n$, $i = 1, \ldots, m$
- cluster into k groups
- minimize sum of squared distances to cluster centers

biconvex program formulation:

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^m z_i^T (\|\bar{x}_1 - x_i\|_2^2, \ldots, \|\bar{x}_k - x_i\|_2^2) \\ \text{subject to} & 0 \preceq z_i \preceq \mathbf{1}, \quad \mathbf{1}^T z_i = 1, \quad i = 1, \ldots, m \end{array}$$

- variables $\bar{x}_1, \ldots, \bar{x}_k \in \mathbf{R}^n$, $z_1, \ldots, z_m \in \mathbf{R}^k$
- $\bar{x}_i$ are the cluster centers
- z_i are the (soft) cluster assignment vectors

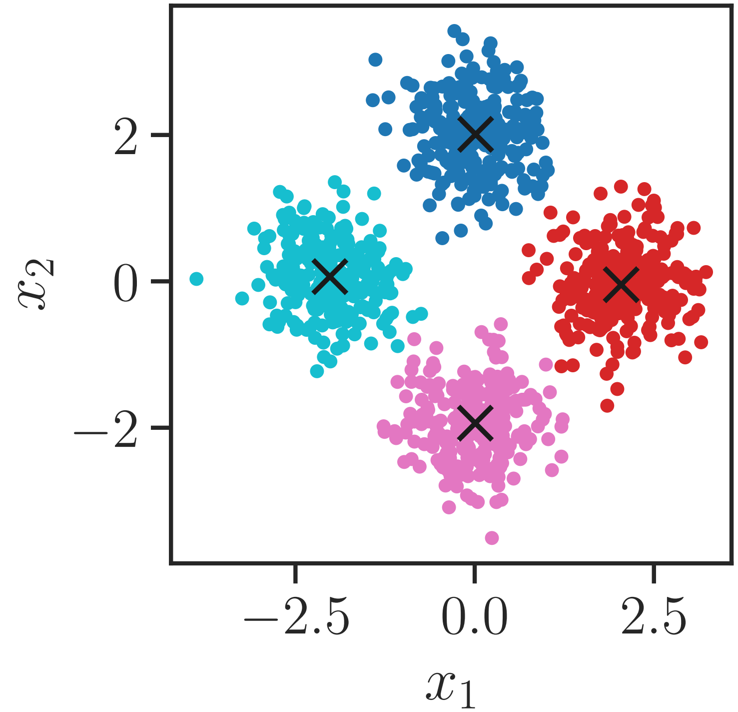
to specify the problem using dbcp:

```
xbars = cp.Variable((k, n))
zs = cp.Variable((m, k), nonneg=True)

obj = cp.sum(cp.multiply(zs, cp.vstack([
    cp.sum(cp.square(xs - c), axis=1) for c in xbars
]).T))
constr = [zs <= 1, cp.sum(zs, axis=1) == 1]
prob = BiconvexProblem(cp.Minimize(obj), [[xbars], [zs]], constr)
```

example:

- $m = 1000$ points in $\mathbf{R}^2$
- $k = 4$ clusters
- ground truth centroids: $(0, 2)$, $(0, -2)$, $(2, 0)$, $(-2, 0)$



Bilinear logistic regression

- given a dataset (X_i, y_i), $i = 1, \ldots, m$, where $X_i \in \mathbf{R}^{n \times k}$ are feature matrices, $y_i \in \{0, 1\}$ are binary labels
- to construct a bilinear classifier

$$\hat{y} = \begin{cases} 1 & \mathbf{tr}(U^T X V) \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

where $U \in \mathbf{R}^{n \times r}$, $V \in \mathbf{R}^{k \times r}$ are the classifier coefficients

maximum likelihood estimation problem:

$$\text{maximize} \quad \textstyle\sum_{i=1}^{m} y_i \, \mathbf{tr}(U^T X_i V) - \log(1 + \exp(\mathbf{tr}(U^T X_i V)))$$

is biconvex in U and V

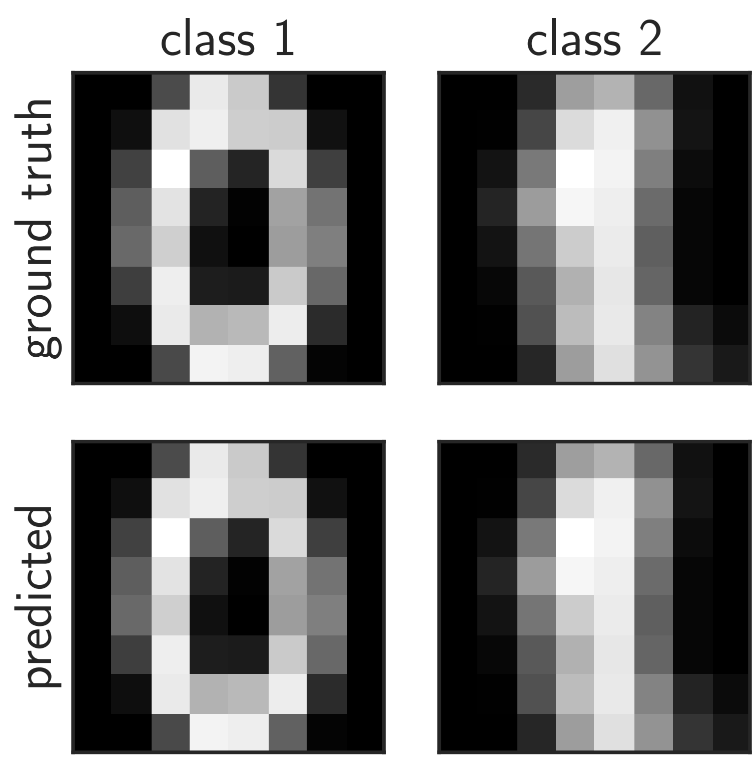
to specify the problem using `dbcp`:

```
U = cp.Variable((n, r))
V = cp.Variable((k, r))

obj = 0
for X, y in zip(Xs, ys):
    obj += cp.sum(cp.multiply(y, cp.trace(U.T @ X @ V))
            - cp.logistic(cp.trace(U.T @ X @ V)))
prob = BiconvexProblem(cp.Maximize(obj), [[U], [V]])
```

example:

- $m = 360$
  handwritten digits of ‘0’ and ‘1’
- 8×8 pixel images ($n = k = 8$)
- rank $r = 2$



Sparse dictionary learning

- given data matrix $Y \in \mathbf{R}^{m \times n}$
- find dictionary $D \in \mathbf{R}^{m \times k}$ and sparse code $X \in \mathbf{R}^{k \times n}$ s.t. $DX \approx Y$

biconvex program formulation:

$$\begin{array}{ll} \text{minimize} & \|DX - Y\|_F^2 + \alpha \|X\|_1 \\ \text{subject to} & \|D\|_F \leq \beta \end{array}$$

variables D, X; hyperparameters $\alpha, \beta > 0$; $\|\cdot\|_1$ componentwise ℓ_1-norm

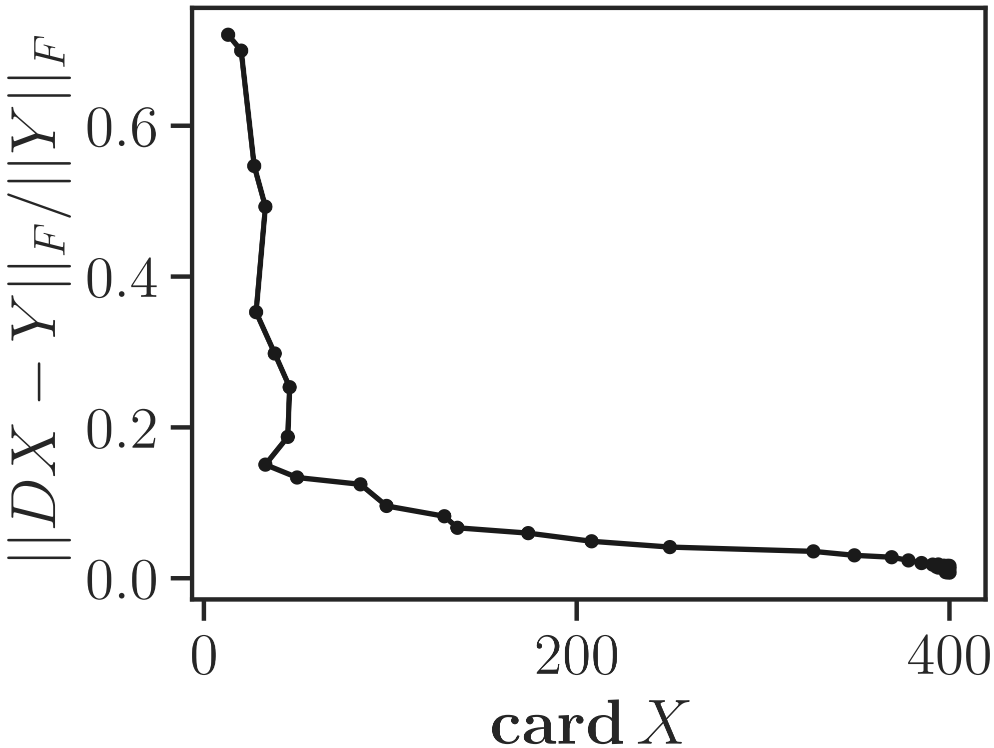
to specify the problem using `dbcp`:

```
D = cp.Variable((m, k))
X = cp.Variable((k, n))

obj = cp.Minimize(cp.sum_squares(D @ X - Y) + alpha * cp.norm1(X))
prob = BiconvexProblem(obj, [[D], [X]], [cp.norm(D,'fro') <= beta])
```

example:

- $m = 10$, $n = 20$, $k = 20$, $\beta = 1$
- $Y \in \mathbf{R}^{m \times n}$ generated from standard normal distribution
- problem solved with different α from 10^{-2} to 1



Input-output hidden Markov model (IO-HMM)

given dataset $\{(x(t), y(t))\}_{t=1}^m$ generated from a K-state IO-HMM:

- $x(t) \in \mathbf{R}^n$ are the features, $y(t) \in \{0, 1\}$ are the labels
- $\hat{z}(t) \in \{1, \ldots, K\}$: hidden state labels, modeled as a Markov chain

$$\hat{z}(t) \sim \begin{cases} \mathrm{Cat}(p_{\mathrm{init}}) & t = 0 \\ \mathrm{Cat}(p_{\hat{z}(t-1)}) & t > 0 \end{cases}$$

  - $p_{\mathrm{init}} \in \mathbf{R}^K$ $(\mathbf{1}^T p_{\mathrm{init}} = 1)$: initial state distribution
  - $P_{\mathrm{tr}} \in \mathbf{R}^{K \times K}$ $(P_{\mathrm{tr}}\mathbf{1} = \mathbf{1})$: state transition matrix
  - $p_{\hat{z}(t-1)} \in \mathbf{R}^K$ is the $\hat{z}(t-1)$th row of P_{tr}
- response $y(t)$ generated from a logistic model:

$$\mathbf{prob}(y(t) = 1) = 1/(1 + \exp(-x(t)^T \theta_{\hat{z}(t)}))$$

  - $\theta_{\hat{z}(t)} \in \{\theta_1, \ldots, \theta_K\} \subseteq \mathbf{R}^n$ is the coefficient

model fitting problem: estimate parameters $p_{\mathrm{init}}, P_{\mathrm{tr}}, \theta_1, \ldots, \theta_K$ by maximizing the data log-likelihood:

$$
\begin{array}{ll}
\text{minimize} & -\sum_{t=1}^{m} z(t)^T \Big(y(t)x(t)^T\theta_k - \log(1+\exp(x(t)^T\theta_k)) \Big)_{k=1}^{K} \\
& \qquad +\alpha_\theta \sum_{k=1}^{K} \|\theta_k\|_2^2 + \alpha_z \sum_{t=1}^{m-1} D_{\mathrm{kl}}(z(t), z(t+1)) \\
\text{subject to} & 0 \preceq z(t) \preceq \mathbf{1}, \quad \mathbf{1}^T z(t) = 1, \quad t = 1, \ldots, m \\
& \theta_k \in \mathcal{C}_k, \quad k = 1, \ldots, K
\end{array}
$$

- variables $z(t) \in \mathbf{R}^K$, $\theta_k \in \mathbf{R}^n$
- hyperparameters $\alpha_\theta, \alpha_z > 0$
- D_{kl} is the Kullback-Leibler divergence
- $\mathcal{C}_k$ are convex sets specifying prior knowledge on θ_k

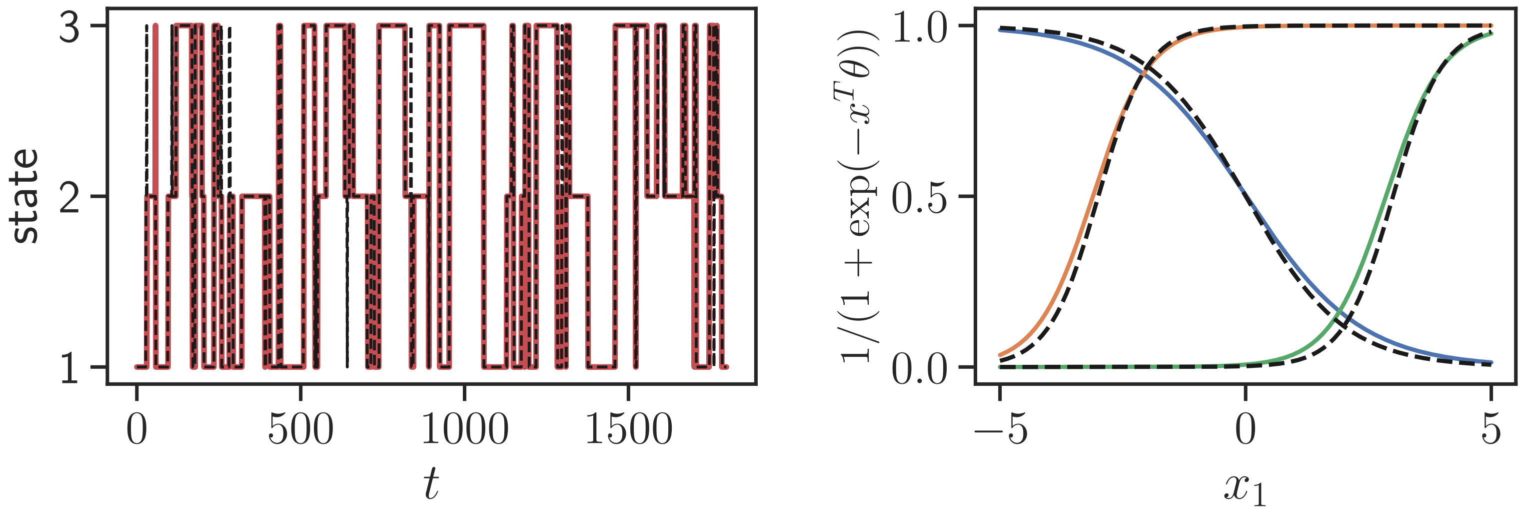
example:

- $m = 1800$, $n = 2$, $K = 3$, $P_{\mathrm{tr}} = \begin{bmatrix} 0.95 & 0.025 & 0.025 \\ 0.025 & 0.95 & 0.025 \\ 0.025 & 0.025 & 0.95 \end{bmatrix}$
- $x(t) \sim (\mathcal{U}(-5, 5),\ 1)$
- constraint sets $\mathcal{C}_k$ given by: $\theta_{1,1} \leq 0, \quad \theta_{2,1} \geq 0, \quad \theta_{3,1} \geq 0, \quad \theta_{2,2} \geq \theta_{3,2}$

```
thetas = cp.Variable((K, n))
zs = cp.Variable((m, K), nonneg=True)

rs = [-cp.multiply(ys, xs @ thetas[k]) + cp.logistic(xs @ thetas[k])
    for k in range(K)]
obj = cp.Minimize(
    cp.sum(cp.multiply(zs, cp.vstack(rs).T))
    + alpha_theta * cp.sum_squares(thetas)
    + alpha_z * cp.sum(cp.kl_div(zs[:-1], zs[1:])))
constr = [
    thetas[0][0] <= 0,
    thetas[1][0] >= 0,
    thetas[2][0] >= 0,
    thetas[1][1] >= thetas[2][1],
    zs <= 1, cp.sum(zs, axis=1) == 1
]

prob = BiconvexRelaxProblem(obj, ([zs], [thetas]), constr)
```



- ground truth shown in black dashed lines

4. Summary

- summary

- resources

Summary

DBCP:

- modeling framework for biconvex optimization problems
- specify biconvex problems in a human readable way, close to the math
- enable fast experiment and prototyping of different problem structures
- fully open-sourced package `dbcp`, integrated with CVXPY ecosystem

Resources

- paper:

  Disciplined biconvex programming.
  H. Zhu and J. Boedecker. *arXiv:2511.01813*, November 2025.

- implementation: Python package `dbcp`

  `https://github.com/nrgrp/dbcp`

- examples and tutorial:

  `https://github.com/nrgrp/dbcp/tree/main/examples`

- this slides deck and many other useful materials:

  `https://haozhu10015.github.io`

5. Appendices

- DBCP product ruleset
- Generalized inequality constraints

DBCP product ruleset

most DBCP rules are inherited from DCP, with additional product rules:

1. A valid DBCP convex product expression should include variables in both the left-hand and right-hand expressions, and should be one of the following forms:

$$\begin{gathered} \textit{affine} * \textit{affine} \\ \textit{affine-nonneg} * \textit{convex} \quad \text{or} \quad \textit{affine-nonpos} * \textit{concave} \\ \textit{convex-nonneg} * \textit{convex-nonneg} \quad \text{or} \quad \textit{concave-nonpos} * \textit{concave-nonpos} \end{gathered}$$

2. There exists no loop in the variable interaction graph of the overall expression, where the edge between two variables indicates that they appear on different sides in a product expression as described in the above rule.

Generalized inequality constraints

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

- variables $x \in \mathcal{X}$, $y \in \mathcal{Y}$; f_0 biconvex; h_i biaffine
- $f_i \colon \mathcal{X} \times \mathcal{Y} \to \mathbf{R}^{q_i}$ biconvex; inequality w.r.t. proper cones $\mathcal{K}_i \subseteq \mathbf{R}^{q_i}$

feasibility problem (relaxation):

$$\begin{array}{ll} \text{minimize} & \mathbf{1}^T s + \|t\|_1 \\ \text{subject to} & s \succeq 0 \\ & f_i(x,y) \preceq_{\mathcal{K}_i} s_i e_{\mathcal{K}_i}, \quad i=1,\ldots,m \\ & h_i(x,y) = t_i, \quad i=1,\ldots,p, \end{array}$$

- $e_{\mathcal{K}_i} \succeq_{\mathcal{K}_i} 0$ is any positive element of cone $\mathcal{K}_i$
- examples: $(0,1) \in \mathbf{R}^q$ for SOC in $\mathbf{R}^q$; $I \in \mathbf{R}^{n\times n}$ for PSD cone $\mathbf{S}^n_+$

References

[BV04] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[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.

[ZB25] H. Zhu and J. Boedecker. Disciplined biconvex programming. *arXiv Preprint arXiv:2511.01813*, 2025.