Probabilistic Graphical Models

Prof. Joschka Boedecker and Hao Zhu

Department of Computer Science
University of Freiburg

universität freiburg

2. Bayesian classifiers

- Probabilistic classification
  - Probabilistic classification problems
  - Naive Bayesian classifiers
  - Augmented Bayesian classifiers
  - Semi-naive Bayesian classifiers

- Multi-label classification
  - Multi-dimensional classification problems
  - Basic approaches
  - Chain classifiers

Outline

- Probabilistic classification
  - Probabilistic classification problems
  - Naive Bayesian classifiers
  - Augmented Bayesian classifiers
  - Semi-naive Bayesian classifiers

- Multi-label classification
  - Multi-dimensional classification problems
  - Basic approaches
  - Chain classifiers

Probabilistic classification problems

given a set of samples X and a set of class labels Y $(\mathbf{dom}(Y) \subseteq \mathbf{Z}_+)$

- 'ordinary' classifier: $f\colon X \to Y$
- probabilistic classifier:

$$f(x) = (\ldots, \mathbf{P}(y_i \mid x), \ldots), \quad i = 1, \ldots, \mathbf{card}(Y)$$

  - $f(x)^T \mathbf{1} = 1$
  - $\hat{y} = \mathrm{argmax}_y\, \mathbf{P}(y \mid x)$

Probabilistic classification problems

Bayesian approach

$$\mathbf{P}(y \mid x) = \frac{\mathbf{P}(x \mid y)\mathbf{P}(y)}{\mathbf{P}(x)}$$

- $\mathbf{P}(x)$: normalizing constant independent of labels
- $\mathbf{P}(y)$: prior on class labels
- $\mathbf{P}(x \mid y)$: likelihood of sample x under label y

$$\begin{aligned} \mathbf{P}(x \mid y) &= \mathbf{P}(x_1, \ldots, x_n \mid y) \\ &= \mathbf{P}(x_1 \mid y)\mathbf{P}(x_2 \mid x_1, y) \cdots \mathbf{P}(x_n \mid x_{n-1}, \ldots, x_1, y) \end{aligned}$$

  - can be difficult to calculate

Naive Bayesian classifiers

assumption: $x_1, \ldots, x_n$ are independent given y

$$\mathbf{P}(y \mid x) = \frac{\mathbf{P}(x \mid y)\mathbf{P}(y)}{\mathbf{P}(x)} \propto \mathbf{P}(y)\mathbf{P}(x_1, \ldots, x_n \mid y) = \mathbf{P}(y) \prod_{i=1}^{n} \mathbf{P}(x_i \mid y)$$

parameter learning

- prior $\mathbf{P}(y)$:

$$\mathbf{P}(y_i) = \frac{1}{\mathbf{card}(Y)} \quad \text{or} \quad \mathbf{P}(y_i) = \frac{\#\text{ samples in class } y_i}{\#\text{ samples in total}}$$

- likelihood $\mathbf{P}(x \mid y)$:

$$\mathbf{P}(x_k \mid y_i) = \frac{\#\text{ samples in class } y_i \text{ with feature } x_k}{\#\text{ samples in class } y_i}$$

for all $k = 1, \ldots, n$, $y_i \in Y$

Naive Bayesian classifiers

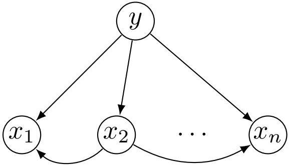
handling continuous features: Gaussian naive Bayes

$$p(x_k \mid y_i) = \frac{1}{\sqrt{2\pi}\sigma_{k|y_i}} \exp\left(-\frac{\left(x_k - \mu_{k|y_i}\right)^2}{2\sigma^2_{k|y_i}}\right), \quad k = 1, \ldots, n$$

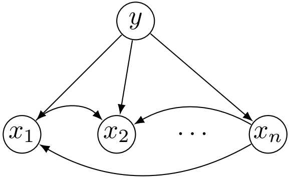
- $\mu_{k|y_i} = \mathbf{E}[X_k \mid y_i]$, $k = 1, \ldots, n$
- $\sigma_{k|y_i} = \sqrt{\mathbf{var}(X_k \mid y_i)} = \sqrt{\mathbf{E}\left[\left(X_k - \mathbf{E}[X_k \mid y_i]\right)^2 \,\middle|\, y_i\right]}$, $k = 1, \ldots, n$
- $\mathbf{P}(x_k \mid y_i) \propto p(x_k \mid y_i)$

Augmented Bayesian classifiers

assumption: some dependency structure (tree, DAG, ...) exists between $x_1, \ldots, x_n$ given y



tree augmented Bayesian classifiers



Bayesian network augmented Bayesian classifiers

parameter learning

$$\mathbf{P}(x \mid y) = \mathbf{P}(x_1, \ldots, x_n \mid y) = \prod_{i=1}^{n} \mathbf{P}(x_i \mid \mathbf{pa}(x_i), y)$$

Semi-naive Bayesian classifiers

basic idea: naive Bayes + feature selection

- eliminate or join interdependent features given the class label

feature selection metrics

- local measure: e.g., mutual information
- global measure: e.g., performance of the classifier with and without the feature

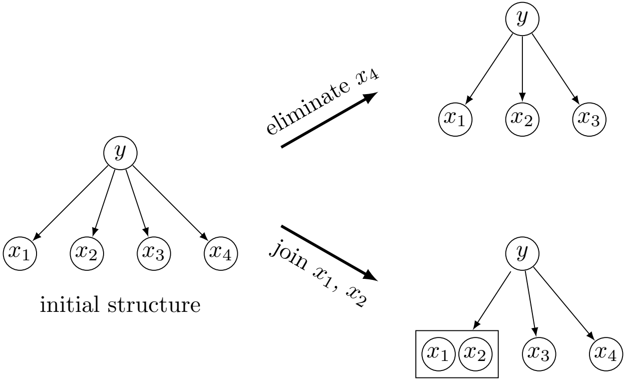
model structure learning process

- bottom-up: start from an empty structure and add features
- top-down: from a full structure with all the features and eliminate (or combine) features

parameter learning: the same as naive Bayesian classifiers

Semi-naive Bayesian classifiers

example: top-down structure learning



Outline

- Probabilistic classification
  - Probabilistic classification problems
  - Naive Bayesian classifiers
  - Augmented Bayesian classifiers
  - Semi-naive Bayesian classifiers

- Multi-label classification
  - Multi-dimensional classification problems
  - Basic approaches
  - Chain classifiers

Multi-dimensional classification problems

given a set of samples X and a set of class labels Y $(\mathbf{dom}(Y) \subseteq \mathbf{Z}_+^m)$

probabilistic classifier f:

$$f(x) = (\ldots, \mathbf{P}(y \mid x), \ldots) = (\ldots, \mathbf{P}(y_1, \ldots, y_m \mid x), \ldots)$$

- $y \in Y$ is a m-dimensional vector
- $f(x)^T \mathbf{1} = 1$
- $\hat{y} = \operatorname{argmax}_y \mathbf{P}(y \mid x)$

multi-label classification: $\mathbf{dom}(Y_i) = \{0, 1\}, i = 1, \ldots, m$

Basic approaches

binary relevance

- assumption: no dependencies between all pairs of classes
- solve m independent binary classification problems
- a classifier is independently learnt for each class $Y_1, \ldots, Y_m$
- final prediction is a simple concatenation of results from all classifier, $\hat{y} = (\hat{y}_1, \ldots, \hat{y}_m)$

label power-set

- basic idea: transform multi-label classification to single-class scenario
- define a mapping $g\colon Y \to Y'$ from $\mathbf{dom}(Y) \subseteq \mathbf{Z}_+^m$ to $\mathbf{dom}(Y') \subseteq \mathbf{Z}_+$
- learn a single-class classifier on Y' given X
- interactions between classes are implicitly considered
- $\mathbf{card}(Y')$ increases exponentially w.r.t. m

Chain classifiers

basic idea: generalize the binary relevance approach to considering some dependencies between classes

- m binary classifiers $(f_1, \ldots, f_m)$ linked in a chain, each corresponding to one class
- the predictions $\hat{y}_1, \ldots, \hat{y}_{i-1}$ from $f_1, \ldots, f_{i-1}$ is incorporated into the features of f_i

$$\hat{y}_1 = \operatorname{argmax}_{y_1} \mathbf{P}(y_1 \mid x)$$
$$\hat{y}_i = \operatorname{argmax}_{y_i} \mathbf{P}(y_i \mid x, \hat{y}_1, \ldots, \hat{y}_{i-1}), \quad i = 2, \ldots, m$$

- model performance depends on the order of classes in the chain

Chain classifiers

circular chain classifier

- $(f_1, \ldots, f_m)$ are connected in a circular way
- the first cycle:

$$\hat{y}_1 = \operatorname{argmax}_{y_1} \mathbf{P}(y_1 \mid x)$$
$$\hat{y}_i = \operatorname{argmax}_{y_i} \mathbf{P}(y_i \mid x, \hat{y}_1, \ldots, \hat{y}_{i-1}), \quad i = 2, \ldots, m$$

- from the second cycle:

$$\hat{y}_i = \operatorname*{argmax}_{y_i} \mathbf{P}(y_i \mid x, \hat{y}_{-i}), \quad i = 1, \ldots, m$$

  each binary classifier in the chain receives the predictions of all other classifiers as additional feature
- repeated for a prefixed number of cycles or until convergence

Chain classifiers

Bayesian chain classifier

- connection between $(f_1, \ldots, f_m)$ represented as a DAG

$$\mathbf{P}(y \mid x) = \mathbf{P}(y_1, \ldots, y_m \mid x) = \prod_{i=1}^{m} \mathbf{P}(y_i \mid \mathbf{pa}(y_i), x)$$

- to get final prediction $\hat{y}$, approximate the hard combinatorial optimization problem

$$\text{maximize (over } y\text{)} \quad \prod_{i=1}^{m} \mathbf{P}(y_i \mid \mathbf{pa}(y_i), x)$$

with a sequence of independent optimization problems

$$\text{maximize (over } y_i\text{)} \quad \mathbf{P}(y_i \mid \mathbf{pa}(y_i), x)$$

for all $i = 1, \ldots, m$

Chain classifiers

example