Lecture 09: Linear Discriminant Analysis

Introduction to Machine Learning [25737]

Sajjad Amini

Sharif University of Technology

Contents

References

Except explicitly cited, the reference for the material in slides is:

- Murphy, K. P. (2022). *Probabilistic machine learning: an introduction.* MIT press.

Section 1

Approach Definitions

Approach Definitions

Discriminant Analysis

Assume we consider the following model for classification:

$$p(y = c|\boldsymbol{x}; \boldsymbol{\theta}) = \frac{p(\boldsymbol{x}|y = c; \boldsymbol{\theta})p(y = c|\boldsymbol{\theta})}{\sum_{c'} p(\boldsymbol{x}|y = c'; \boldsymbol{\theta})p(y = c'|\boldsymbol{\theta})}$$

where:

$p(y = c\|\boldsymbol{\theta})$	Prior probability over labels
$p(\boldsymbol{x}\|y = c; \boldsymbol{\theta})$	Class conditional density

Using special options for class conditional density, we can show that:

$$p(y|\boldsymbol{x}; \boldsymbol{\theta}) = \boldsymbol{w}^T\boldsymbol{x} + \text{constant}$$

The resulting model is known as linear discriminant analysis.

Section 2

Gaussian Discriminant Analysis

Gaussian Discriminant Analysis (GDA)

Class Conditional Density

For Gaussian discriminant analysis, the class conditional density is:

$$p(\boldsymbol{x}|y = c; \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$$

The above selection result in the following posterior over class labels:

$$p(y = c|\boldsymbol{x}, \boldsymbol{\theta}) \propto \pi_c \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$$

where

$$\pi_c = p(y = c|\boldsymbol{\theta})$$

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) = \frac{1}{|2\pi\boldsymbol{\Sigma}_c|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(\boldsymbol{y} - \boldsymbol{\mu}_c)^T \boldsymbol{\Sigma}_c^{-1}(\boldsymbol{y} - \boldsymbol{\mu}_c)\right]$$

Decision Boundary

Quadratic Decision Boundary

Consider the log posterior probability as:

$$\log p(y = c|\boldsymbol{x}; \boldsymbol{\theta}) = \log \pi_c - \frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_c)^T \Sigma_c^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_c) + \text{const}$$

This method is called *Quadratic Discriminant Analysis* (QDA) because the decision boundary is a quadratic function.

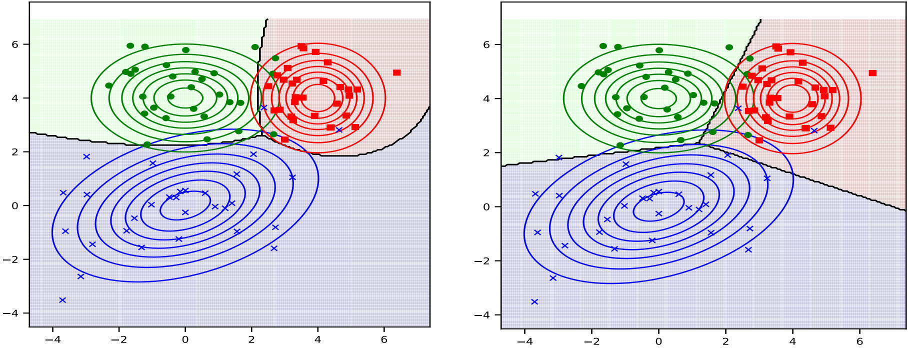
Linear Decision Boundary

If we assume $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_c$, then:

$$\begin{aligned}\log p(y = c|\boldsymbol{x}; \boldsymbol{\theta}) &= \log \pi_c - \frac{1}{2}\boldsymbol{\mu}_c^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_c + \boldsymbol{x}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_c + \text{const} - \frac{1}{2}\boldsymbol{x}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{x} \\ &= \gamma_c + \boldsymbol{x}^T \boldsymbol{\beta}_c + \kappa\end{aligned}$$

This method is called *Linear Discriminant Analysis* (LDA) because the decision boundary is a linear function.

Quadratic vs Linear Discriminant Analysis



(a) QDA

(b) LDA

Figure: Decision boundary comparison

Section 3

Connection Between LDA and MLR

Connection Between LDA and MLR

Similarity

As we can see, LDA can be formulated as:

$$p(y = c|\boldsymbol{x}, \boldsymbol{\theta}) = \frac{\exp(\boldsymbol{\beta}_c^T \boldsymbol{x} + \gamma_c)}{\sum_{c'=1}^{C} \exp(\boldsymbol{\beta}_{c'}^T \boldsymbol{x} + \gamma_{c'})} = \frac{\exp(\boldsymbol{w}_c^T [1; \boldsymbol{x}])}{\sum_{c'=1}^{C} \exp(\boldsymbol{w}_{c'}^T [1; \boldsymbol{x}])}$$

Thus the posterior form is similar to MLR.

Difference

- In LDA, we first estimate prior probability over labels and class conditional density, and derive $\{\boldsymbol{w}_c\}_{c=1}^{C}$ from them.
- In MLR, we estimate $\{\boldsymbol{w}_c\}_{c=1}^{C}$ directly to maximize conditional likelihood $p(y|\boldsymbol{x}, \boldsymbol{\theta})$

MLE

MLE

The likelihood function can be formulated as:

$$p(\mathcal{D}|\boldsymbol{\theta}) = p\left(\{(\boldsymbol{x}_n, y_n)\}_{n=1}^{N}|\boldsymbol{\theta}\right) \overset{(1)}{=} \prod_{n=1}^{N} p(\boldsymbol{x}_n, y_n|\boldsymbol{\theta})$$

$$\overset{(2)}{=} \prod_{n=1}^{N} p(y_n|\boldsymbol{\theta})p(\boldsymbol{x}_n|y_n, \boldsymbol{\theta}) = \prod_{n=1}^{N} \mathrm{Cat}(y_n|\boldsymbol{\pi}) \prod_{c=1}^{C} \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)^{\mathbb{I}(y_n=c)}$$

where we use independency of training samples and probability chain rule for equality (1) and (2), respectively. Note that the parameter vector is:

$$\boldsymbol{\theta} = [\boldsymbol{\pi}; \boldsymbol{\mu}_1; \ldots; \boldsymbol{\mu}_C; \mathrm{vec}(\boldsymbol{\Sigma}_1); \ldots; \mathrm{vec}(\boldsymbol{\Sigma}_C)]$$

MLE (Continue)

The likelihood function and its log version are:

$$p(\mathcal{D}|\boldsymbol{\theta}) = \prod_{n=1}^{N} \mathrm{Cat}(y_n|\boldsymbol{\pi}) \prod_{c=1}^{C} \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)^{\mathbb{I}(y_n=c)}$$

$$\Rightarrow \log\ p(\mathcal{D}|\boldsymbol{\theta}) = \left[\sum_{n=1}^{N}\sum_{c=1}^{C} \mathbb{I}(y_n = c)\log \pi_c\right] + \sum_{c=1}^{C}\left[\sum_{n:y_n=c} \log \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)\right]$$

Using differentiation, we can calculate the model parameters as:

$$\widehat{\pi}_c = \frac{N_c}{N}$$

$$\widehat{\boldsymbol{\mu}}_c = \frac{1}{N_c}\sum_{n:y_n=c} \boldsymbol{x}_n$$

$$\widehat{\boldsymbol{\Sigma}}_c = \frac{1}{N_c}\sum_{n:y_n=c} (\boldsymbol{x}_n - \widehat{\boldsymbol{\mu}}_c)(\boldsymbol{x}_n - \widehat{\boldsymbol{\mu}}_c)^T$$

Tied Covariance Matrices

Tied Covariance Matrices

Tied covariance matrices is the situation where we force all covariance matrices to be equal as:

$$\boldsymbol{\Sigma}_c = \boldsymbol{\Sigma}, \; c = 1, \ldots, C$$

MLE estimation for tied covariance matrix is:

$$\widehat{\boldsymbol{\Sigma}} = \frac{1}{N} \sum_{c=1}^{C} \sum_{n:y_n=c} (\boldsymbol{x}_n - \widehat{\boldsymbol{\mu}}_c)(\boldsymbol{x}_n - \widehat{\boldsymbol{\mu}}_c)^T$$

LDA and Tied Covariance Matrix

When the covariance matrix is tied, QDA simplifies to LDA.

Diagonal LDA

We can simplifies tied covariance matrix further by assumeing it to be diagonal, so: $\boldsymbol{\Sigma}_c = \boldsymbol{D}, \; c = 1, \ldots, C$

Nearest Centroid Classifier

Nearest Centroid Classifier

Assume the prior probability over classes is uniform, so:

$$\pi_c = \frac{1}{C}, \; c = 1, \ldots, C$$

If the covariance matrices are tied, then:

$$\begin{aligned}\widehat{y}(\boldsymbol{x}) &= \underset{c}{\operatorname{argmax}} \;\; \log p(y = c|\boldsymbol{x}, \boldsymbol{\theta}) = \underset{c}{\operatorname{argmin}} \;\; (\boldsymbol{x} - \boldsymbol{\mu}_c)^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_c) \\ &= \underset{c}{\operatorname{argmin}} \;\; \Delta^2_{\boldsymbol{\Sigma}}(\boldsymbol{x}, \boldsymbol{\mu}_c)\end{aligned}$$

Thus the class whose mean has minimum Mahalanobis distance to the query point $\boldsymbol{x}$ is selected as the label.

Section 4

Naive Bayes Classifier

Naive Bayes Classifier

Main Assumption

The input features are mutually independent given the class label. In other words:

$$p(\boldsymbol{x}|y=c,\boldsymbol{\theta}) = \prod_{d=1}^{D} p(x_d|y=c,\boldsymbol{\theta}_{dc})$$

where $\boldsymbol{\theta}_{dc}$ is model parameter vector for conditional density for class c and feature d. The posterior over class label is:

$$p(y=c|\boldsymbol{x},\boldsymbol{\theta}) = \frac{p(y=c|\boldsymbol{\pi})\prod_{d=1}^{D} p(x_d|y=c,\boldsymbol{\theta}_{dc})}{\sum_{c'} p(y=c'|\boldsymbol{\pi})\prod_{d=1}^{D} p(x_d|y=c',\boldsymbol{\theta}_{dc'})}$$

Pros and cons

- The naive model may not hold in many real world application.
- Naive Bayes model is relatively immune to overfitting.

Example Models

Binary Features

In this case $x_d \in \{0, 1\}$ and thus the class conditional density is:

$$p(\boldsymbol{x}|y = c, \boldsymbol{\theta}) = \prod_{d=1}^{D} \mathrm{Ber}(x_d|\theta_{dc})$$

where θ_{dc} shows the probability that $x_d = 1$ in class c. This model is known as *multivariate Bernoulli naive Bayes*.

Categorical Features

In this case $x_d \in \{0, 1, \ldots, K\}$ and thus the class conditional density is:

$$p(\boldsymbol{x}|y = c, \boldsymbol{\theta}) = \prod_{d=1}^{D} \mathrm{Cat}(x_d|\boldsymbol{\theta}_{dc})$$

where θ_{dck} shows the probability that $x_d = k$ in class c.

Example Models

Real-values Features

In this case $x_d \in \mathbb{R}$ and thus we can use univariate Gaussian for each dimension in each class. Thus the class conditional density is:

$$p(\boldsymbol{x}|y=c,\boldsymbol{\theta}) = \prod_{d=1}^{D} \mathcal{N}(x_d|\mu_{dc}, \sigma_{dc}^2)$$

where μ_{dc} and σ_{dc}^2 shows the mean and variance of feature d in class c.

Model Fitting

MLE

The likelihood for the dataset $\mathcal{D}$ is:

$$\begin{aligned} p(\mathcal{D}|\boldsymbol{\theta}) &= \prod_{n=1}^{N} p(y_n|\boldsymbol{\theta}) p(\boldsymbol{x}_n|y_n,\boldsymbol{\theta}) \overset{(1)}{=} \prod_{n=1}^{N} p(y_n|\boldsymbol{\theta}) \prod_{d=1}^{D} p(x_{nd}|y_n,\boldsymbol{\theta}_d) \\ &= \prod_{n=1}^{N} \mathrm{Cat}(y_n|\boldsymbol{\pi}) \prod_{d=1}^{D} \prod_{c=1}^{C} p(x_{nd}|\boldsymbol{\theta}_{dc})^{\mathbb{I}(y_n=c)} \end{aligned}$$

where we use Naive Bayes assumption for equality (1).

Model Fitting

MLE (Continue)

$$\log p(\mathcal{D}|\boldsymbol{\theta}) = \log \prod_{n=1}^{N} \left(\left[\prod_{c=1}^{C} \pi_c^{\mathbb{I}(y_n=c)} \right] \left[\prod_{d=1}^{D} \prod_{c=1}^{C} p(x_{nd}|\boldsymbol{\theta}_{dc})^{\mathbb{I}(y_n=c)} \right] \right)$$

$$= \sum_{n=1}^{N} \left(\log \left[\prod_{c=1}^{C} \pi_c^{\mathbb{I}(y_n=c)} \right] + \log \left[\prod_{d=1}^{D} \prod_{c=1}^{C} p(x_{nd}|\boldsymbol{\theta}_{dc})^{\mathbb{I}(y_n=c)} \right] \right)$$

$$= \sum_{n=1}^{N} \left(\left[\sum_{c=1}^{C} \mathbb{I}(y_n = c) \log \pi_c \right] + \left[\sum_{d=1}^{D} \sum_{c=1}^{C} \mathbb{I}(y_n = c) \log p(x_{nd}|\boldsymbol{\theta}_{dc}) \right] \right)$$

$$= \left[\sum_{n=1}^{N} \sum_{c=1}^{C} \mathbb{I}(y_n = c) \log \pi_c \right] + \left[\sum_{n=1}^{N} \sum_{d=1}^{D} \sum_{c=1}^{C} \mathbb{I}(y_n = c) \log p(x_{nd}|\boldsymbol{\theta}_{dc}) \right]$$

Model Fitting

MLE for $\boldsymbol{\pi}$

Irrespective of class conditional density, the MLE for $\boldsymbol{\pi}$ is the vector of empirical counts as $\widehat{\pi}_c = \frac{N_c}{N}$.

MLE for $\boldsymbol{\theta}_{dc}$

- Binary features: $\widehat{\theta}_{dc} = \frac{N_{dc}}{N_c}$
- Categorical features: $\widehat{\theta}_{dck} = \frac{N_{dck}}{N_c}, \; k = 1, \ldots, K$
- Real-valued features (Univariate Gaussian):

$$\widehat{\mu}_{dc} = \frac{1}{N_c} \sum_{n=1}^{N} \mathbb{I}(y_n = c) x_{nd}$$

$$\widehat{\sigma}_{dc}^2 = \frac{1}{N_c} \sum_{n=1}^{N} \mathbb{I}(y_n = c)(x_{nd} - \widehat{\mu}_{dc})^2$$

Section 5

Generative vs. Discriminative

MLR vs DA

In MLR, we have $p(y|\boldsymbol{x};\boldsymbol{\theta}) = \mathrm{Cat}(y|\mathcal{S}(\boldsymbol{W}^T\boldsymbol{x}+\boldsymbol{b}))$ and the likelihood is:

$$p(\mathcal{D}|\boldsymbol{\theta}) = p(\{y_n\}_{n=1}^N|\{\boldsymbol{x}_n\}_{n=1}^N, \boldsymbol{\theta}) = \prod_{n=1}^N p(y_n|\boldsymbol{x}_n, \boldsymbol{\theta})$$

In Discriminant analysis, we have $p(y=c|\boldsymbol{x};\boldsymbol{\theta}) = \frac{p(\boldsymbol{x}|y=c;\boldsymbol{\theta})p(y=c|\boldsymbol{\theta})}{\sum_{c'} p(\boldsymbol{x}|y=c';\boldsymbol{\theta})p(y=c'|\boldsymbol{\theta})}$ and the likelihood is:

$$p(\mathcal{D}|\boldsymbol{\theta}) = p\left(\{(\boldsymbol{x}_n, y_n)\}_{n=1}^N|\boldsymbol{\theta}\right) = \prod_{n=1}^N p(\boldsymbol{x}_n, y_n|\boldsymbol{\theta}) = \prod_{n=1}^N p(\boldsymbol{x}_n|y_n, \boldsymbol{\theta})p(y_n|\boldsymbol{\theta})$$

Generative vs. Discriminative

Discriminative

By training MLR:

- You have access to $p(y|\boldsymbol{x}, \boldsymbol{\theta})$ which can be used to generate label for a query input $\boldsymbol{x}$ (discriminate the label of $\boldsymbol{x}$).
- You can't generate samples from specific class $y = k$.

Generative

By training DA:

- You have access to $p(y|\boldsymbol{x}, \boldsymbol{\theta})$ which can be used to generate label for a query input $\boldsymbol{x}$ (discriminate the label of $\boldsymbol{x}$).
- You have access to $p(\boldsymbol{x}|y, \boldsymbol{\theta})$ that can be used to generate samples from specific class $y = k$.