These are the slides of the lecture

Pattern Recognition
Winter term 2020/21
Friedrich-Alexander University of Erlangen-Nuremberg.

Erlangen, January 8, 2021
Prof. Dr.-Ing. Andreas Maier

Pattern Recognition (PR)

Prof. Dr.-Ing. Andreas Maier
Pattern Recognition Lab (CS 5), Friedrich-Alexander-Universität Erlangen-Nürnberg
Winter Term 2020/21

Discriminant Analysis I

Discriminant Analysis

Discriminant analysis methods are *discriminative modeling* methods that model the posterior through its factorization

$$p(y|\boldsymbol{x}) = \frac{p(y) \cdot p(\boldsymbol{x}|y)}{\sum_y p(y) \cdot p(\boldsymbol{x}|y)}$$

Gaussian Classifier

We call the Bayesian classifier Gaussian, if the class conditional density $p(\boldsymbol{x}|y)$ is Gaussian, i. e.

$$\begin{aligned} p(\boldsymbol{x}|y) &= \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y) \\ &= \frac{1}{\sqrt{\det 2\pi \boldsymbol{\Sigma}_y}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_y)^T \boldsymbol{\Sigma}_y^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_y)} \end{aligned}$$

where

$\boldsymbol{x} \in \mathbb{R}^d$: d-dimensional feature vector

$\boldsymbol{\mu}_y \in \mathbb{R}^d$: mean vector of class y

$\boldsymbol{\Sigma}_y \in \mathbb{R}^{d \times d}$: positive definite covariance matrix.

Gaussian Classifier (cont.)

Facts about Gaussian classifiers:

- In general the decision boundary is quadratic in the components x_i of the feature vector $\boldsymbol{x}$.
- If all classes share the same covariance, the decision boundary is linear in the components x_i of the feature vector $\boldsymbol{x}$.
- If all covariance matrices are diagonal matrices, then we get a Naïve Bayes classifier.

Gaussian Classifier (cont.)

Facts about Gaussian classifiers (cont.):

- If the joint covariance matrix is $\boldsymbol{\Sigma}$ and priors are identical, classification requires the minimization of the Mahalanobis distance

$$y^* = \operatorname*{argmin}_y \frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_y)^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_y)$$

Gaussian Classifier (cont.)

Facts about Gaussian classifiers (cont.):

- If the joint covariance matrix is $\boldsymbol{\Sigma}$ and priors are identical, classification requires the minimization of the Mahalanobis distance

$$y^* = \underset{y}{\operatorname{argmin}} \frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_y)^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_y)$$

- If all covariance matrices are the identity matrix, we get the Nearest Neighbor classifier based on the L_2-norm:

$$y^* = \underset{y}{\operatorname{argmin}} \frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_y)^T(\boldsymbol{x} - \boldsymbol{\mu}_y)$$

  The prototype vectors are the mean vectors.

Gaussian Classifier (cont.)

From linear to quadratic decision boundaries:

A compromise between linear and quadratic decision boundaries can be achieved by using regularized covariance matrices:

$$\boldsymbol{\Sigma}_y(\alpha) = \alpha \boldsymbol{\Sigma}_y + (1-\alpha)\boldsymbol{\Sigma}$$

where $\alpha \in [0,1]$ and $\boldsymbol{\Sigma}$ denotes the joint covariance.

Obviously we have the extremes:

- Linear decision boundary: $\alpha = 0$
- Quadratic decision boundary: $\alpha = 1$

Feature Transform

Can we find a feature transform

$$\phi : \mathbb{R}^d \quad \rightarrow \quad \mathbb{R}^d$$

to generate features $\phi(\boldsymbol{x})$ that share the same covariance matrix?

Feature Transform (cont.)

The symmetric positive semidefinite covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{d \times d}$ can be decomposed using SVD:

$$\boldsymbol{\Sigma} \quad =$$

Feature Transform (cont.)

The symmetric positive semidefinite covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{d \times d}$ can be decomposed using SVD:

$$\boldsymbol{\Sigma} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T$$

Feature Transform (cont.)

The symmetric positive semidefinite covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{d \times d}$ can be decomposed using SVD:

$$\boldsymbol{\Sigma} \quad = \quad \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T = (\boldsymbol{U}\boldsymbol{D}^{\frac{1}{2}})(\boldsymbol{U}\boldsymbol{D}^{\frac{1}{2}})^T$$

Feature Transform (cont.)

The symmetric positive semidefinite covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{d \times d}$ can be decomposed using SVD:

$$\boldsymbol{\Sigma} \quad = \quad \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T = (\boldsymbol{U}\boldsymbol{D}^{\frac{1}{2}})(\boldsymbol{U}\boldsymbol{D}^{\frac{1}{2}})^T = (\boldsymbol{U}\boldsymbol{D}^{\frac{1}{2}}) \cdot \boldsymbol{I} \cdot (\boldsymbol{U}\boldsymbol{D}^{\frac{1}{2}})^T$$

where $\boldsymbol{I} \in \mathbb{R}^{d \times d}$ is the identity matrix.

Feature Transform (cont.)

The symmetric positive semidefinite covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{d \times d}$ can be decomposed using SVD:

$$\boldsymbol{\Sigma} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T = (\boldsymbol{U}\boldsymbol{D}^{\frac{1}{2}})(\boldsymbol{U}\boldsymbol{D}^{\frac{1}{2}})^T = (\boldsymbol{U}\boldsymbol{D}^{\frac{1}{2}}) \cdot \boldsymbol{I} \cdot (\boldsymbol{U}\boldsymbol{D}^{\frac{1}{2}})^T$$

where $\boldsymbol{I} \in \mathbb{R}^{d \times d}$ is the identity matrix.

- Determinant:

$$\det \boldsymbol{\Sigma} = \prod_{i=1}^{d} d_{i,i},$$

where $d_{i,i}$ are the diagonal elements of $\boldsymbol{D}$, i. e. the singular values.

Feature Transform (cont.)

The symmetric positive semidefinite covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{d \times d}$ can be decomposed using SVD:

$$\boldsymbol{\Sigma} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T = (\boldsymbol{U}\boldsymbol{D}^{\frac{1}{2}})(\boldsymbol{U}\boldsymbol{D}^{\frac{1}{2}})^T = (\boldsymbol{U}\boldsymbol{D}^{\frac{1}{2}}) \cdot \boldsymbol{I} \cdot (\boldsymbol{U}\boldsymbol{D}^{\frac{1}{2}})^T$$

where $\boldsymbol{I} \in \mathbb{R}^{d \times d}$ is the identity matrix.

- Determinant:

$$\det \boldsymbol{\Sigma} = \prod_{i=1}^{d} d_{i,i},$$

  where $d_{i,i}$ are the diagonal elements of $\boldsymbol{D}$, i. e. the singular values.

- Inverse:

$$\boldsymbol{\Sigma}^{-1} = \boldsymbol{U}\boldsymbol{D}^{-1}\boldsymbol{U}^T = (\boldsymbol{U}\boldsymbol{D}^{-\frac{1}{2}}) \cdot \boldsymbol{I} \cdot (\boldsymbol{U}\boldsymbol{D}^{-\frac{1}{2}})^T$$

Feature Transform (cont.)

Now we incorporate this:

$$\mathcal{N}(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{\sqrt{\det 2\pi\boldsymbol{\Sigma}}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}$$

Feature Transform (cont.)

Now we incorporate this:

$$
\begin{aligned}
\mathcal{N}(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma}) &= \frac{1}{\sqrt{\det 2\pi\boldsymbol{\Sigma}}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})} \\
&= \frac{1}{\sqrt{\det 2\pi\boldsymbol{\Sigma}}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T (\boldsymbol{U}\boldsymbol{D}^{-\frac{1}{2}})\cdot\boldsymbol{I}\cdot(\boldsymbol{U}\boldsymbol{D}^{-\frac{1}{2}})^T(\boldsymbol{x}-\boldsymbol{\mu})}
\end{aligned}
$$

Feature Transform (cont.)

Now we incorporate this:

$$
\begin{aligned}
\mathcal{N}(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma}) &= \frac{1}{\sqrt{\det 2\pi\boldsymbol{\Sigma}}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})} \\
&= \frac{1}{\sqrt{\det 2\pi\boldsymbol{\Sigma}}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T(\boldsymbol{U}\boldsymbol{D}^{-\frac{1}{2}})\cdot\boldsymbol{I}\cdot(\boldsymbol{U}\boldsymbol{D}^{-\frac{1}{2}})^T(\boldsymbol{x}-\boldsymbol{\mu})} \\
&= \frac{1}{\sqrt{\det 2\pi\boldsymbol{\Sigma}}} e^{-\frac{1}{2}\left((\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{U}^T)\boldsymbol{x}-(\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{U}^T)\boldsymbol{\mu}\right)^T\boldsymbol{I}\left((\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{U}^T)\boldsymbol{x}-(\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{U}^T)\boldsymbol{\mu}\right)}
\end{aligned}
$$

Feature Transform (cont.)

The classwise transform ϕ_y is even a linear function:

$$\boldsymbol{x}' = \phi_y(\boldsymbol{x}) = \boldsymbol{D}_y^{-\frac{1}{2}} \boldsymbol{U}_y^T \boldsymbol{x}$$

Feature Transform (cont.)

The classwise transform ϕ_y is even a linear function:

$$\boldsymbol{x}' = \phi_y(\boldsymbol{x}) = \boldsymbol{D}_y^{-\frac{1}{2}} \boldsymbol{U}_y^T \boldsymbol{x}$$

It is straight forward to show that $\boldsymbol{x}'$ is normally distributed

$$p(\boldsymbol{x}'|y) = \mathcal{N}(\boldsymbol{x}'; \boldsymbol{\mu}'_y, \Sigma'_y) = \mathcal{N}(\boldsymbol{x}'; \boldsymbol{D}_y^{-\frac{1}{2}} \boldsymbol{U}_y^T \boldsymbol{\mu}_y, \boldsymbol{I})$$

Feature Transform (cont.)

Conclusions:

- All classes y share the same covariance matrix that is the identity matrix.
- The decision boundary is linear.

Feature Transform (cont.)

Conclusions:

- All classes y share the same covariance matrix that is the identity matrix.
- The decision boundary is linear.
- ⚠ Huge disadvantage:
  feature transform depends on class number y!
- If we have a classified training set, we can compute a transform for each class such that all covariance matrices are the identity matrix.
- Classification requires the application of different transforms.

Pattern Recognition Lab

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

TECHNISCHE FAKULTÄT

Next Time in

Pattern Recognition

Linear Discriminant Analysis

Input: training data: $S = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), (\boldsymbol{x}_3, y_3), \ldots, (\boldsymbol{x}_m, y_m)\}$

Linear Discriminant Analysis

Input: training data: $S = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), (\boldsymbol{x}_3, y_3), \ldots, (\boldsymbol{x}_m, y_m)\}$

1. ML estimation of the joint covariance matrix:

$$\widehat{\boldsymbol{\Sigma}} = \frac{1}{m} \sum_{i=1}^{m} (\boldsymbol{x}_i - \boldsymbol{\mu}_{y_i})(\boldsymbol{x}_i - \boldsymbol{\mu}_{y_i})^T$$

Linear Discriminant Analysis

Input: training data: $S = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), (\boldsymbol{x}_3, y_3), \ldots, (\boldsymbol{x}_m, y_m)\}$

1. ML estimation of the joint covariance matrix:

$$\widehat{\boldsymbol{\Sigma}} = \frac{1}{m} \sum_{i=1}^{m} (\boldsymbol{x}_i - \boldsymbol{\mu}_{y_i})(\boldsymbol{x}_i - \boldsymbol{\mu}_{y_i})^T$$

2. Compute SVD of covariance matrix: $\widehat{\boldsymbol{\Sigma}} = \boldsymbol{UDU}^T$

Linear Discriminant Analysis

Input: training data: $S = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), (\boldsymbol{x}_3, y_3), \ldots, (\boldsymbol{x}_m, y_m)\}$

1. ML estimation of the joint covariance matrix:

$$\widehat{\boldsymbol{\Sigma}} = \frac{1}{m} \sum_{i=1}^{m} (\boldsymbol{x}_i - \boldsymbol{\mu}_{y_i})(\boldsymbol{x}_i - \boldsymbol{\mu}_{y_i})^T$$

2. Compute SVD of covariance matrix: $\widehat{\boldsymbol{\Sigma}} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T$
3. Assign transform:

$$\phi = \boldsymbol{D}^{-\frac{1}{2}} \boldsymbol{U}^T$$

Linear Discriminant Analysis

Input: training data: $S = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), (\boldsymbol{x}_3, y_3), \ldots, (\boldsymbol{x}_m, y_m)\}$

1. ML estimation of the joint covariance matrix:

$$\widehat{\Sigma} = \frac{1}{m} \sum_{i=1}^{m} (\boldsymbol{x}_i - \boldsymbol{\mu}_{y_i})(\boldsymbol{x}_i - \boldsymbol{\mu}_{y_i})^T$$

2. Compute SVD of covariance matrix: $\widehat{\Sigma} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T$
3. Assign transform:

$$\phi = \boldsymbol{D}^{-\frac{1}{2}} \boldsymbol{U}^T$$

4. Compute mean vectors for all y

$$\boldsymbol{\mu}'_y = \phi(\boldsymbol{\mu}_y) = \boldsymbol{D}^{-\frac{1}{2}} \boldsymbol{U}^T \boldsymbol{\mu}_y$$

Linear Discriminant Analysis

Input: training data: $S = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), (\boldsymbol{x}_3, y_3), \ldots, (\boldsymbol{x}_m, y_m)\}$

1. ML estimation of the joint covariance matrix:

$$\widehat{\Sigma} = \frac{1}{m} \sum_{i=1}^{m} (\boldsymbol{x}_i - \boldsymbol{\mu}_{y_i})(\boldsymbol{x}_i - \boldsymbol{\mu}_{y_i})^T$$

2. Compute SVD of covariance matrix: $\widehat{\Sigma} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T$
3. Assign transform:

$$\phi = \boldsymbol{D}^{-\frac{1}{2}} \boldsymbol{U}^T$$

4. Compute mean vectors for all y

$$\boldsymbol{\mu}'_y = \phi(\boldsymbol{\mu}_y) = \boldsymbol{D}^{-\frac{1}{2}} \boldsymbol{U}^T \boldsymbol{\mu}_y$$

Output: feature transform ϕ, transformed mean vectors $\boldsymbol{\mu}'_y$

Linear Discriminant Analysis (cont.)

Decision rule using sphered data $\phi(\boldsymbol{x})$:

$$y^* =$$

Linear Discriminant Analysis (cont.)

Decision rule using sphered data $\phi(\boldsymbol{x})$:

$$y^* = \underset{y}{\operatorname{argmax}}\, p(y|\phi(\boldsymbol{x}))$$

Linear Discriminant Analysis (cont.)

Decision rule using sphered data $\phi(\boldsymbol{x})$:

$$
\begin{aligned}
y^* &= \underset{y}{\operatorname{argmax}}\, p(y|\phi(\boldsymbol{x})) \\
&= \underset{y}{\operatorname{argmax}} \left\{ \log p(y) - \frac{1}{2}(\phi(\boldsymbol{x}) - \phi(\boldsymbol{\mu}_y))^T (\phi(\boldsymbol{x}) - \phi(\boldsymbol{\mu}_y)) \right\}
\end{aligned}
$$

Linear Discriminant Analysis (cont.)

Decision rule using sphered data $\phi(\boldsymbol{x})$:

$$
\begin{aligned}
y^* &= \underset{y}{\operatorname{argmax}}\, p(y|\phi(\boldsymbol{x})) \\
&= \underset{y}{\operatorname{argmax}} \left\{ \log p(y) - \frac{1}{2}(\phi(\boldsymbol{x}) - \phi(\boldsymbol{\mu}_y))^T (\phi(\boldsymbol{x}) - \phi(\boldsymbol{\mu}_y)) \right\} \\
&= \underset{y}{\operatorname{argmin}} \left\{ \frac{1}{2} \|\phi(\boldsymbol{x}) - \phi(\boldsymbol{\mu}_y)\|_2^2 - \log p(y) \right\}
\end{aligned}
$$

where $\|.\|_2$ denotes the L_2 norm.

Linear Discriminant Analysis (cont.)

Conclusions:

- If all classes share the same prior,
  the decision rule is the Nearest Neighbor decision rule,
  where transformed mean vectors serve as prototypes.

- The feature transform ϕ does not change the dimension of features.

Linear Discriminant Analysis (cont.)

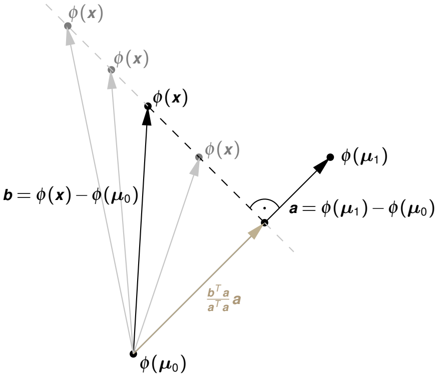

Fig.: Nearest Neighbor classification for two classes

Linear Discriminant Analysis (cont.)

2 classes: insights from geometrical analysis of sphered data

- Angle between $\phi(\boldsymbol{x})$ and $(\phi(\boldsymbol{\mu}_1) - \phi(\boldsymbol{\mu}_0))$ can be used for decision making.

- Decision rule:

$$y^* = \begin{cases} 0, & \text{if} \quad \phi(\boldsymbol{x})^T(\phi(\boldsymbol{\mu}_1) - \phi(\boldsymbol{\mu}_0)) < \frac{1}{2}(\phi(\boldsymbol{\mu}_1)^T\phi(\boldsymbol{\mu}_1) - \phi(\boldsymbol{\mu}_0)^T\phi(\boldsymbol{\mu}_0)) \\ 1, & \text{otherwise.} \end{cases}$$

- Coordinate orthogonal to the 1-D subspace spanned by $(\phi(\boldsymbol{\mu}_1) - \phi(\boldsymbol{\mu}_0))$ does not affect relative distances.

Linear Discriminant Analysis (cont.)

K classes: insights from geometrical analysis of sphered data

- Class centroids span $(K-1)$-dimensional subspace.
- Relative differences are not affected by coordinates in the $(d-K+1)$-dimensional subspace that is orthogonal to the $(K-1)$-dimensional subspace spanned by class centroids.

Linear Discriminant Analysis (cont.)

Objective:

Will we gain an advantage if we transform features by

$$\phi : \mathbb{R}^d \to \mathbb{R}^k$$

in higher $(k > d)$ or lower dimensional $(k < d)$ spaces?

Lessons Learned

- Relationship between Bayesian classifier, Gaussian classifier, and Nearest Neighbor classifier.
- Mahalanobis distance
- Linear Discriminant Analysis is a regularized Nearest Neighbor classifier
- Class centroids span $(K-1)$-dimensional subspace

Pattern Recognition Lab

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

TECHNISCHE FAKULTÄT

Next Time in

Pattern Recognition

Further Readings

You are required to be familiar with linear algebra and matrix calculus:

- SIAMS best selling book in the last decade:

  Lloyd N. Trefethen, David Bau III:
  Numerical Linear Algebra,
  SIAM, Philadelphia, 1997.

- All about matrix derivatives and related problems is described in the Matrix Cookbook: `http://www.matrixcookbook.com`

Basics on discriminant analysis can be found in

- T. Hastie, R. Tibshirani, and J. Friedman:
  The Elements of Statistical Learning –
  Data Mining, Inference, and Prediction,
  2nd edition, Springer, New York, 2009.

Comprehensive Questions

- What is a Gaussian classifier?
- What is the idea behind the feature transform for the LDA?
- Formulate the LDA for normally distributed classes.
- What is the dimensionality of the LDA subspace for K classes?