These are the slides of the lecture

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

Please feel free to reuse any of the figures and slides, as long as you keep a reference to the source of these slides at `https://lme.tf.fau.de/teaching/` acknowledging the authors Niemann, Hornegger, Hahn, Steidl, Nöth, Seitz, Rodriguez, Das and Maier.

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 II

Rank Reduced Linear Discriminant Analysis

Problem: How to choose an L-dimensional subspace with $L = K - 1$ that is good for LDA?

Idea: Maximize the spread of the L-dimensional projection of centroids.

Solution: Principal component analysis, i. e.
we compute the principal components of the covariance matrix of the mean vectors

$$\boldsymbol{\mu}'_y = \phi(\boldsymbol{\mu}_y) \in \mathbb{R}^{K-1},$$

where $y = 1, 2, \ldots, K$.

Rank Reduced Linear Discriminant Analysis (cont.)

In Principle Component Analysis (PCA) we compute a linear mapping $\mathbf{\Phi} \in \mathbb{R}^{L\times(K-1)}$ that results in the highest spread of projected features:

Rank Reduced Linear Discriminant Analysis (cont.)

In Principle Component Analysis (PCA) we compute a linear mapping $\mathbf{\Phi} \in \mathbb{R}^{L \times (K-1)}$ that results in the highest spread of projected features:

$$\mathbf{\Phi}^* = \underset{\mathbf{\Phi}}{\operatorname{argmax}} \left(\frac{1}{K} \sum_{y=1}^{K} (\mathbf{\Phi}\boldsymbol{\mu}'_y - \mathbf{\Phi}\bar{\boldsymbol{\mu}}')^T (\mathbf{\Phi}\boldsymbol{\mu}'_y - \mathbf{\Phi}\bar{\boldsymbol{\mu}}') + \sum_{i=1}^{L} \lambda_i (\|\mathbf{\Phi}_i\|_2^2 - 1) \right)$$

where we applied the Lagrange multiplier method to allow for the maximization of the spread subject to

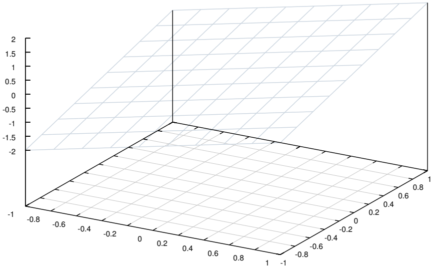
$$\|\mathbf{\Phi}_i\|_2^2 = 1 \,, \quad i = 1, \ldots, K-1 \,.$$

Here $\|\mathbf{\Phi}_i\|_2^2$ denotes the L_2 norm of the i-th row vector of $\mathbf{\Phi}$.

Excursus: Optimization with Constraints

Lagrange Multipliers: simple example

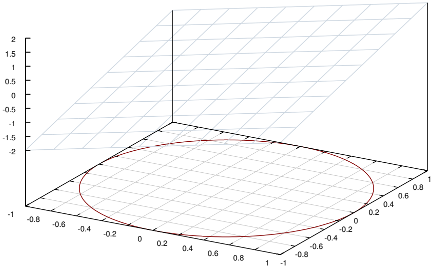
- Find the maximum of $f(x,y) = x + y$



Excursus: Optimization with Constraints

Lagrange Multipliers: simple example

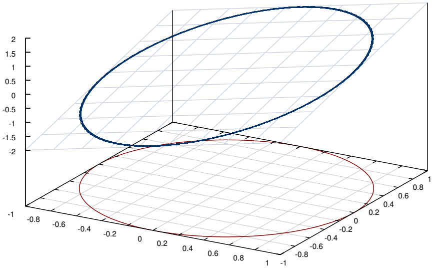
- Find the maximum of $f(x,y) = x + y$
- constraint: $x^2 + y^2 = 1$



Excursus: Optimization with Constraints

Lagrange Multipliers: simple example

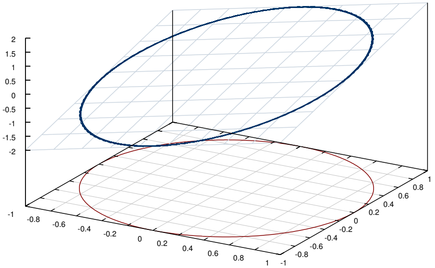
- Find the maximum of $f(x,y) = x + y$
- constraint: $x^2 + y^2 = 1$



Excursus: Optimization with Constraints

Lagrange Multipliers: simple example

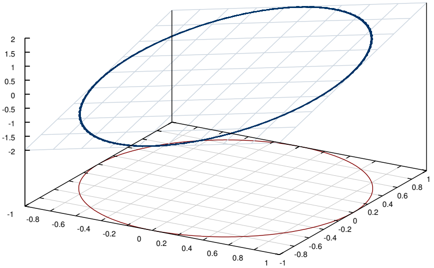
- Find the maximum of $f(x,y) = x + y$
- constraint: $x^2 + y^2 = 1$
- Lagrange function: $L(x,y,\lambda) = x + y + \lambda(x^2 + y^2 - 1)$



Excursus: Optimization with Constraints

Lagrange Multipliers: simple example

- Find the maximum of
  $f(x,y) = x + y$
- constraint: $x^2 + y^2 = 1$
- Lagrange function:
  $L(x,y,\lambda) = x + y + \lambda(x^2 + y^2 - 1)$
- Set the partial derivatives to zero:
  $\frac{\partial L}{\partial x} = \frac{\partial L}{\partial y} = \frac{\partial L}{\partial \lambda} \overset{!}{=} 0$



Rank Reduced Linear Discriminant Analysis (cont.)

We need some facts from matrix calculus:
☞ www.matrixcookbook.com[1]

1. Let $\boldsymbol{\mu}$ denote the mean and $\boldsymbol{\Sigma}$ the covariance matrix of a random vector $\boldsymbol{x}$, then we get:

$$E[(\boldsymbol{A}\boldsymbol{x})^T(\boldsymbol{A}\boldsymbol{x})] = \operatorname{tr}(\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^T) + (\boldsymbol{A}\boldsymbol{\mu})^T(\boldsymbol{A}\boldsymbol{\mu})$$

[1] website currently off-line – you can still download it ☞ here

Rank Reduced Linear Discriminant Analysis (cont.)

We need some facts from matrix calculus:
☞ www.matrixcookbook.com[1]

1. Let $\boldsymbol{\mu}$ denote the mean and $\boldsymbol{\Sigma}$ the covariance matrix of a random vector $\boldsymbol{x}$, then we get:

$$E[(\boldsymbol{A}\boldsymbol{x})^T(\boldsymbol{A}\boldsymbol{x})] = \operatorname{tr}(\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^T) + (\boldsymbol{A}\boldsymbol{\mu})^T(\boldsymbol{A}\boldsymbol{\mu})$$

2. The matrix derivative is:

$$\frac{\partial \operatorname{tr}(\boldsymbol{X}\boldsymbol{B}\boldsymbol{X}^T)}{\partial \boldsymbol{X}} = \boldsymbol{X}\boldsymbol{B}^T + \boldsymbol{X}\boldsymbol{B}$$

[1] website currently off-line – you can still download it ☞ here

Rank Reduced Linear Discriminant Analysis (cont.)

For our optimization problem this implies:

$$\frac{\partial}{\partial \boldsymbol{\Phi}} \left\{ \frac{1}{K} \sum_{y=1}^{K} (\boldsymbol{\Phi}\boldsymbol{\mu}'_y - \boldsymbol{\Phi}\bar{\boldsymbol{\mu}}')^T (\boldsymbol{\Phi}\boldsymbol{\mu}'_y - \boldsymbol{\Phi}\bar{\boldsymbol{\mu}}') + \sum_{i=1}^{L} \lambda_i (\|\boldsymbol{\Phi}_i\|_2^2 - 1) \right\}$$

Rank Reduced Linear Discriminant Analysis (cont.)

For our optimization problem this implies:

$$\frac{\partial}{\partial \boldsymbol{\Phi}} \left\{ \frac{1}{K} \sum_{y=1}^{K} (\boldsymbol{\Phi}\boldsymbol{\mu}'_y - \boldsymbol{\Phi}\bar{\boldsymbol{\mu}}')^T (\boldsymbol{\Phi}\boldsymbol{\mu}'_y - \boldsymbol{\Phi}\bar{\boldsymbol{\mu}}') + \sum_{i=1}^{L} \lambda_i (\|\boldsymbol{\Phi}_i\|_2^2 - 1) \right\}$$

$$= \quad \frac{\partial}{\partial \boldsymbol{\Phi}} \left\{ \frac{1}{K} \sum_{y=1}^{K} \left(\boldsymbol{\Phi}(\boldsymbol{\mu}'_y - \bar{\boldsymbol{\mu}}')\right)^T \left(\boldsymbol{\Phi}(\boldsymbol{\mu}'_y - \bar{\boldsymbol{\mu}}')\right) + \sum_{i=1}^{L} \lambda_i (\|\boldsymbol{\Phi}_i\|_2^2 - 1) \right\}$$

Rank Reduced Linear Discriminant Analysis (cont.)

For our optimization problem this implies:

$$\frac{\partial}{\partial \boldsymbol{\Phi}} \left\{ \frac{1}{K} \sum_{y=1}^{K} (\boldsymbol{\Phi}\boldsymbol{\mu}'_y - \boldsymbol{\Phi}\bar{\boldsymbol{\mu}}')^T (\boldsymbol{\Phi}\boldsymbol{\mu}'_y - \boldsymbol{\Phi}\bar{\boldsymbol{\mu}}') + \sum_{i=1}^{L} \lambda_i (\|\boldsymbol{\Phi}_i\|_2^2 - 1) \right\}$$

$$= \quad \frac{\partial}{\partial \boldsymbol{\Phi}} \left\{ \frac{1}{K} \sum_{y=1}^{K} \left(\boldsymbol{\Phi}(\boldsymbol{\mu}'_y - \bar{\boldsymbol{\mu}}') \right)^T \left(\boldsymbol{\Phi}(\boldsymbol{\mu}'_y - \bar{\boldsymbol{\mu}}') \right) + \sum_{i=1}^{L} \lambda_i (\|\boldsymbol{\Phi}_i\|_2^2 - 1) \right\}$$

$$= \quad \frac{\partial}{\partial \boldsymbol{\Phi}} \left\{ \mathrm{tr}(\boldsymbol{\Phi}\, \boldsymbol{\Sigma}_{\text{inter}}\, \boldsymbol{\Phi}^T) + \sum_{i=1}^{L} \lambda_i (\|\boldsymbol{\Phi}_i\|_2^2 - 1) \right\} \stackrel{!}{=} 0.$$

Rank Reduced Linear Discriminant Analysis (cont.)

Now we compute the partial derivatives:

$$\frac{\partial}{\partial \boldsymbol{\Phi}} \left\{ \mathrm{tr}(\boldsymbol{\Phi}\, \boldsymbol{\Sigma}_{\text{inter}}\, \boldsymbol{\Phi}^T) + \sum_{i=1}^{L} \lambda_i (\|\boldsymbol{\Phi}_i\|_2^2 - 1) \right\} = 2\boldsymbol{\Phi}\boldsymbol{\Sigma}_{\text{inter}} + 2\boldsymbol{\lambda}\boldsymbol{\Phi} = 0$$

Rank Reduced Linear Discriminant Analysis (cont.)

Now we compute the partial derivatives:

$$\frac{\partial}{\partial \boldsymbol{\Phi}} \left\{ \operatorname{tr}(\boldsymbol{\Phi}\, \boldsymbol{\Sigma}_{\text{inter}}\, \boldsymbol{\Phi}^T) + \sum_{i=1}^{L} \lambda_i (\|\boldsymbol{\Phi}_i\|_2^2 - 1) \right\} = 2\boldsymbol{\Phi}\boldsymbol{\Sigma}_{\text{inter}} + 2\boldsymbol{\lambda}\boldsymbol{\Phi} = 0$$

This results in the eigenvalue and eigenvector problem:

$$\boldsymbol{\Sigma}_{\text{inter}} \boldsymbol{\Phi}^T = \boldsymbol{\lambda}' \boldsymbol{\Phi}^T$$

Rank Reduced Linear Discriminant Analysis (cont.)

Now we compute the partial derivatives:

$$\frac{\partial}{\partial \mathbf{\Phi}} \left\{ \mathrm{tr}(\mathbf{\Phi}\, \mathbf{\Sigma}_{\text{inter}}\, \mathbf{\Phi}^T) + \sum_{i=1}^{L} \lambda_i (\|\mathbf{\Phi}_i\|_2^2 - 1) \right\} = 2\mathbf{\Phi}\mathbf{\Sigma}_{\text{inter}} + 2\boldsymbol{\lambda}\mathbf{\Phi} = 0$$

This results in the eigenvalue and eigenvector problem:

$$\mathbf{\Sigma}_{\text{inter}} \mathbf{\Phi}^T = \boldsymbol{\lambda}' \mathbf{\Phi}^T$$

Note:

In original PCA, the transform $\mathbf{\Phi}$ maximizes the overall spread using the covariance matrix of all features:

$$\mathbf{\Sigma} \mathbf{\Phi}^T = \boldsymbol{\lambda}' \mathbf{\Phi}^T$$

Rank Reduced Linear Discriminant Analysis (cont.)

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. Compute the covariance matrix of transformed mean vectors

$$\widehat{\Sigma}_{\text{inter}} = \frac{1}{K} \sum_{y=1}^{K} (\boldsymbol{\mu}'_y - \bar{\boldsymbol{\mu}}')(\boldsymbol{\mu}'_y - \bar{\boldsymbol{\mu}}')^T,$$

where $\bar{\boldsymbol{\mu}}' = \frac{1}{K} \cdot \sum_{y=1}^{K} \boldsymbol{\mu}'_y$.

Rank Reduced Linear Discriminant Analysis (cont.)

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. Compute the covariance matrix of transformed mean vectors

$$\widehat{\boldsymbol{\Sigma}}_{\text{inter}} = \frac{1}{K} \sum_{y=1}^{K} (\boldsymbol{\mu}'_y - \bar{\boldsymbol{\mu}}')(\boldsymbol{\mu}'_y - \bar{\boldsymbol{\mu}}')^T,$$

   where $\bar{\boldsymbol{\mu}}' = \frac{1}{K} \cdot \sum_{y=1}^{K} \boldsymbol{\mu}'_y$.

2. Compute the L eigenvectors of the covariance matrix belonging to the largest eigenvalues.

Rank Reduced Linear Discriminant Analysis (cont.)

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. Compute the covariance matrix of transformed mean vectors

$$\widehat{\boldsymbol{\Sigma}}_{\text{inter}} = \frac{1}{K} \sum_{y=1}^{K} (\boldsymbol{\mu}'_y - \bar{\boldsymbol{\mu}}')(\boldsymbol{\mu}'_y - \bar{\boldsymbol{\mu}}')^T,$$

   where $\bar{\boldsymbol{\mu}}' = \frac{1}{K} \cdot \sum_{y=1}^{K} \boldsymbol{\mu}'_y$.

2. Compute the L eigenvectors of the covariance matrix belonging to the largest eigenvalues.

3. The eigenvectors are the rows of the mapping $\boldsymbol{\Phi}$ from the $(K-1)$- to the L-dimensional feature space.

Output: matrix $\boldsymbol{\Phi}$

Pattern Recognition Lab

FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG

TECHNISCHE FAKULTÄT

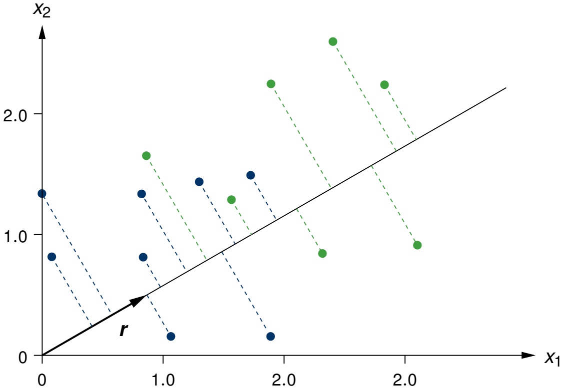
Next Time in

Pattern Recognition

Fisher Transform

The described method to compute the LDA mapping is not the original derivation.

Original method



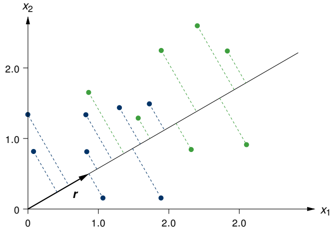
Fisher Transform

The described method to compute the LDA mapping is not the original derivation.

Original method

- Project samples $\boldsymbol{x}_i$ onto a straight line with direction $\boldsymbol{r}$, $\|\boldsymbol{r}\|_2 = 1$:

$$\tilde{x}_i = \boldsymbol{x}_i^T \boldsymbol{r}$$



Fisher Transform

The described method to compute the LDA mapping is not the original derivation.

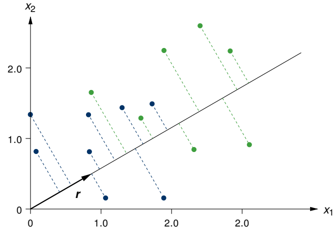
Original method

- Project samples $\boldsymbol{x}_i$ onto a straight line with direction $\boldsymbol{r}$, $\|\boldsymbol{r}\|_2 = 1$:

$$\tilde{x}_i = \boldsymbol{x}_i^T \boldsymbol{r}$$

- Maximize the ratio of the between-class scatter and the within-class scatter:

$$\boldsymbol{r}^* = \underset{\boldsymbol{r}}{\operatorname{argmax}}\, J(\boldsymbol{r}) = \underset{\boldsymbol{r}}{\operatorname{argmax}} \frac{|\tilde{\mu}_1 - \tilde{\mu}_2|^2}{\tilde{s}_1^2 + \tilde{s}_2^2}$$

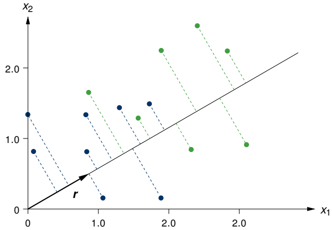


Fisher Transform

The described method to compute the LDA mapping is not the original derivation.

Original method

- Project samples $\boldsymbol{x}_i$ onto a straight line with direction $\boldsymbol{r}$, $\|\boldsymbol{r}\|_2 = 1$:
$$\tilde{x}_i = \boldsymbol{x}_i^T \boldsymbol{r}$$
- Maximize the ratio of the between-class scatter and the within-class scatter:
$$\boldsymbol{r}^* = \underset{\boldsymbol{r}}{\operatorname{argmax}}\, J(\boldsymbol{r}) = \underset{\boldsymbol{r}}{\operatorname{argmax}} \frac{|\tilde{\mu}_1 - \tilde{\mu}_2|^2}{\tilde{s}_1^2 + \tilde{s}_2^2}$$
- Classify by applying a threshold to $\tilde{x}_i$



Fisher Transform (cont.)

Finding $\boldsymbol{r}^*$

1. Mean and scatter matrix for each class:

$$\boldsymbol{\mu}_k = \frac{1}{m_k} \sum_{\substack{i=1 \\ y_i=k}}^{m_k} \boldsymbol{x}_i$$

$$\boldsymbol{S}_k = \sum_{\substack{i=1 \\ y_i=k}}^{m_k} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T$$

2. Within-class scatter matrix:

$$\boldsymbol{S}_W = \boldsymbol{S}_1 + \boldsymbol{S}_2$$

3. Between-class scatter matrix:

$$\boldsymbol{S}_B = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T$$

Fisher Transform (cont.)

Finding $\boldsymbol{r}^*$

4. Expressing $\tilde{\mu}_k$ and $\tilde{s}_k^2$ of the projected samples in terms of $\boldsymbol{\mu}_k$ and $\boldsymbol{S}_k$:

$$\tilde{\mu}_k = \frac{1}{m_k} \sum_{\substack{i=1 \\ y_i=k}}^{m_k} \tilde{x}_i$$

$$\tilde{s}_k^2 = \sum_{\substack{i=1 \\ y_i=k}}^{m_k} (\tilde{x}_i - \tilde{\mu}_k)^2$$

Fisher Transform (cont.)

Finding $\boldsymbol{r}^*$

4. Expressing $\tilde{\mu}_k$ and $\tilde{s}_k^2$ of the projected samples in terms of $\boldsymbol{\mu}_k$ and $\boldsymbol{S}_k$:

$$
\begin{aligned}
\tilde{\mu}_k &= \frac{1}{m_k} \sum_{\substack{i=1 \\ y_i=k}}^{m_k} \tilde{x}_i = \frac{1}{m_k} \sum_{\substack{i=1 \\ y_i=k}}^{m_k} \boldsymbol{r}^T \boldsymbol{x}_i \\
\tilde{s}_k^2 &= \sum_{\substack{i=1 \\ y_i=k}}^{m_k} (\tilde{x}_i - \tilde{\mu}_k)^2
\end{aligned}
$$

Fisher Transform (cont.)

Finding $\boldsymbol{r}^*$

4. Expressing $\tilde{\mu}_k$ and $\tilde{s}_k^2$ of the projected samples in terms of $\boldsymbol{\mu}_k$ and $\boldsymbol{S}_k$:

$$
\begin{aligned}
\tilde{\mu}_k &= \frac{1}{m_k} \sum_{\substack{i=1 \\ y_i=k}}^{m_k} \tilde{x}_i = \frac{1}{m_k} \sum_{\substack{i=1 \\ y_i=k}}^{m_k} \boldsymbol{r}^T \boldsymbol{x}_i = \boldsymbol{r}^T \boldsymbol{\mu}_k \\
\tilde{s}_k^2 &= \sum_{\substack{i=1 \\ y_i=k}}^{m_k} (\tilde{x}_i - \tilde{\mu}_k)^2
\end{aligned}
$$

Fisher Transform (cont.)

Finding $\boldsymbol{r}^*$

4. Expressing $\tilde{\mu}_k$ and $\tilde{s}_k^2$ of the projected samples in terms of $\boldsymbol{\mu}_k$ and $\boldsymbol{S}_k$:

$$\begin{aligned}
\tilde{\mu}_k &= \frac{1}{m_k} \sum_{\substack{i=1 \\ y_i=k}}^{m_k} \tilde{x}_i = \frac{1}{m_k} \sum_{\substack{i=1 \\ y_i=k}}^{m_k} \boldsymbol{r}^T \boldsymbol{x}_i = \boldsymbol{r}^T \boldsymbol{\mu}_k \\
\tilde{s}_k^2 &= \sum_{\substack{i=1 \\ y_i=k}}^{m_k} (\tilde{x}_i - \tilde{\mu}_k)^2 = \sum_{\substack{i=1 \\ y_i=k}}^{m_k} (\boldsymbol{r}^T \boldsymbol{x}_i - \boldsymbol{r}^T \boldsymbol{\mu}_k)^2
\end{aligned}$$

Fisher Transform (cont.)

Finding $\boldsymbol{r}^*$

4. Expressing $\tilde{\mu}_k$ and $\tilde{s}_k^2$ of the projected samples in terms of $\boldsymbol{\mu}_k$ and $\boldsymbol{S}_k$:

$$\tilde{\mu}_k = \frac{1}{m_k}\sum_{\substack{i=1\\y_i=k}}^{m_k}\tilde{x}_i = \frac{1}{m_k}\sum_{\substack{i=1\\y_i=k}}^{m_k}\boldsymbol{r}^T\boldsymbol{x}_i = \boldsymbol{r}^T\boldsymbol{\mu}_k$$

$$\tilde{s}_k^2 = \sum_{\substack{i=1\\y_i=k}}^{m_k}(\tilde{x}_i - \tilde{\mu}_k)^2 = \sum_{\substack{i=1\\y_i=k}}^{m_k}(\boldsymbol{r}^T\boldsymbol{x}_i - \boldsymbol{r}^T\boldsymbol{\mu}_k)^2 = \boldsymbol{r}^T\boldsymbol{S}_k\boldsymbol{r}$$

Fisher Transform (cont.)

Finding $\boldsymbol{r}^*$

4. Expressing $\tilde{\mu}_k$ and $\tilde{s}_k^2$ of the projected samples in terms of $\boldsymbol{\mu}_k$ and $\boldsymbol{S}_k$:

$$\begin{aligned}
\tilde{\mu}_k &= \frac{1}{m_k} \sum_{\substack{i=1\\ y_i=k}}^{m_k} \tilde{x}_i = \frac{1}{m_k} \sum_{\substack{i=1\\ y_i=k}}^{m_k} \boldsymbol{r}^T \boldsymbol{x}_i = \boldsymbol{r}^T \boldsymbol{\mu}_k \\
\tilde{s}_k^2 &= \sum_{\substack{i=1\\ y_i=k}}^{m_k} (\tilde{x}_i - \tilde{\mu}_k)^2 = \sum_{\substack{i=1\\ y_i=k}}^{m_k} (\boldsymbol{r}^T \boldsymbol{x}_i - \boldsymbol{r}^T \boldsymbol{\mu}_k)^2 = \boldsymbol{r}^T \boldsymbol{S}_k \boldsymbol{r}
\end{aligned}$$

5. Plug it into $J(\boldsymbol{r})$:

$$J(\boldsymbol{r}) = \frac{|\tilde{\mu}_1 - \tilde{\mu}_2|^2}{\tilde{s}_1^2 + \tilde{s}_2^2}$$

Fisher Transform (cont.)

Finding $\boldsymbol{r}^*$

4. Expressing $\tilde{\mu}_k$ and $\tilde{s}_k^2$ of the projected samples in terms of $\boldsymbol{\mu}_k$ and $\boldsymbol{S}_k$:

$$\begin{aligned}\tilde{\mu}_k &= \frac{1}{m_k}\sum_{\substack{i=1\\y_i=k}}^{m_k}\tilde{x}_i = \frac{1}{m_k}\sum_{\substack{i=1\\y_i=k}}^{m_k}\boldsymbol{r}^T\boldsymbol{x}_i = \boldsymbol{r}^T\boldsymbol{\mu}_k\\ \tilde{s}_k^2 &= \sum_{\substack{i=1\\y_i=k}}^{m_k}(\tilde{x}_i-\tilde{\mu}_k)^2 = \sum_{\substack{i=1\\y_i=k}}^{m_k}(\boldsymbol{r}^T\boldsymbol{x}_i-\boldsymbol{r}^T\boldsymbol{\mu}_k)^2 = \boldsymbol{r}^T\boldsymbol{S}_k\boldsymbol{r}\end{aligned}$$

5. Plug it into $J(\boldsymbol{r})$:

$$J(\boldsymbol{r}) = \frac{|\tilde{\mu}_1-\tilde{\mu}_2|^2}{\tilde{s}_1^2+\tilde{s}_2^2} = \frac{\boldsymbol{r}^T\boldsymbol{S}_B\boldsymbol{r}}{\boldsymbol{r}^T\boldsymbol{S}_W\boldsymbol{r}}$$

This is known as the Generalized Rayleigh Quotient.

Fisher Transform (cont.)

Finding $\boldsymbol{r}^*$

6. Maximizing the Generalized Rayleigh Quotient is equivalent to solving the following generalized eigenvalue problem:

$$\begin{aligned} \boldsymbol{S}_\mathrm{B}\boldsymbol{r}^* &= \lambda \boldsymbol{S}_\mathrm{W}\boldsymbol{r}^* \\ \boldsymbol{S}_\mathrm{W}^{-1}\boldsymbol{S}_\mathrm{B}\boldsymbol{r}^* &= \lambda \boldsymbol{r}^* \end{aligned}$$

7. Note: $\boldsymbol{S}_\mathrm{B}\boldsymbol{r}^*$ is always in the direction of $\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2$;
   no need to compute the eigenvalues and eigenvectors of $\boldsymbol{S}_\mathrm{W}^{-1}\boldsymbol{S}_\mathrm{B}$!

   The direction of $\boldsymbol{r}^*$ is:

$$\boldsymbol{r}^* = \boldsymbol{S}_\mathrm{W}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$$

Fisher Transform (cont.)

- Usually the total linear mapping for LDA is computed dimension by dimension through the maximization of the Rayleigh ratio for each projection axis $\boldsymbol{a}^*$:

$$\boldsymbol{a}^* = \underset{\boldsymbol{a}}{\operatorname{argmax}} \frac{\boldsymbol{a}^T \Sigma_{\text{inter}} \boldsymbol{a}}{\boldsymbol{a}^T \Sigma_{\text{intra}} \boldsymbol{a}}$$

Fisher Transform (cont.)

- Usually the total linear mapping for LDA is computed dimension by dimension through the maximization of the Rayleigh ratio for each projection axis $\boldsymbol{a}^*$:

$$\boldsymbol{a}^* = \underset{\boldsymbol{a}}{\operatorname{argmax}} \frac{\boldsymbol{a}^T \boldsymbol{\Sigma}_{\text{inter}} \boldsymbol{a}}{\boldsymbol{a}^T \boldsymbol{\Sigma}_{\text{intra}} \boldsymbol{a}}$$

- The solution is a generalized eigenvalue problem: $\boldsymbol{a}$ is the eigenvector of

$$\boldsymbol{\Sigma}_{\text{intra}}^{-1} \boldsymbol{\Sigma}_{\text{inter}}$$

that belongs to the largest eigenvalue.

Fisher Transform (cont.)

In literature the optimization problem is mostly rewritten:

- Equivalent constrained optimization problem

$$\begin{aligned} \text{maximize:} \quad & \boldsymbol{r}^T \boldsymbol{\Sigma}_{\text{inter}} \boldsymbol{r} \\ \text{subject to:} \quad & \boldsymbol{r}^T \boldsymbol{\Sigma}_{\text{intra}} \boldsymbol{r} = 1 \end{aligned}$$

- Lagrange multiplier method ($\lambda > 0$):

$$\boldsymbol{r}^* = \underset{\boldsymbol{r}}{\operatorname{argmax}} \left\{ \boldsymbol{r}^T \boldsymbol{\Sigma}_{\text{inter}} \boldsymbol{r} - \lambda \boldsymbol{r}^T \boldsymbol{\Sigma}_{\text{intra}} \boldsymbol{r} \right\}$$

Dimensionality Reduction

A few comments on dimensionality reduction:

- PCA does not require a classified set of feature vectors (in contrast to LDA).
- PCA transformed features are approximately normally distributed (central limit theorem).
- Components of PCA transformed features are mutually independent.
- There exist many other methods for dimensionality reduction, e. g., Sammon transform, independent component analysis.
- Usually the estimation of transforms is computationally prohibited.
- Johnson-Lindenstrauss lemma: If vectors are projected onto a randomly selected subspace of suitably high dimension, then the distances between the vectors are approximately preserved.

Pattern Recognition Lab

FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG

TECHNISCHE FAKULTÄT

Next Time in

Pattern Recognition

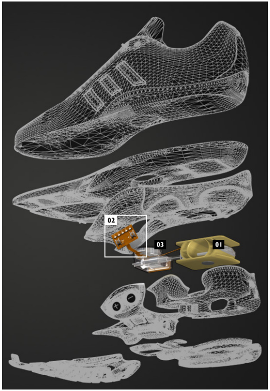
The adidas_1: A Digital Revolution in Sports

- For the first time ever, sport specific information can be processed with a running shoe
- A built-in microprocessor permits an adaptation of the shoe to the prevailing run situation
  - Running speed
  - Runner fatigue
  - Running surface
  - ...
- Pattern Recognition at the LME provides the algorithms used for recognition

The adidas_1: System Overview

Important parts of the adidas_1:

- A cushioning element (01) with a magnetic system for compression measurement
  - $f_{\text{sample}} = 1\text{kHz}$
  - resolution $\Delta d = 0.1\,\text{mm}$
- A microcontroller and user interface (02)
  - $f_{\text{clock}} = 24\,\text{MHz}$
  - 8 kB program memory
- A motor for cushioning adaptation using a cable system (03)



The adidas_1: Classification Framework Requirements

- Only a few, simple features can be calculated in real time
- The classification system has to be efficient,
  but computationally undemanding
- LDA classifier yields a linear decision boundary and can be implemented using a polynomial of order one with weights α_i and features x_i
- In the two class case:

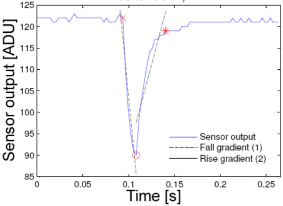
$$\operatorname{sgn}(\boldsymbol{\alpha}^T \boldsymbol{x} + \alpha_0) = \operatorname{sgn}\left(\alpha_1 x_1 + \alpha_2 x_2 + \ldots + \alpha_d x_d + \alpha_0\right)$$

  yields decision for either class.

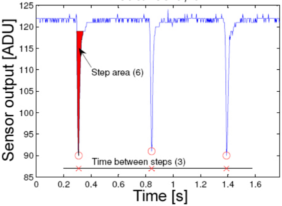
Classification System: Computed Features

- 19 features initially computed for classification experiments
- Feature selection: 3 features selected for implementation

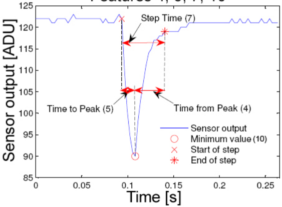
Features 1, 2



Features 3, 6



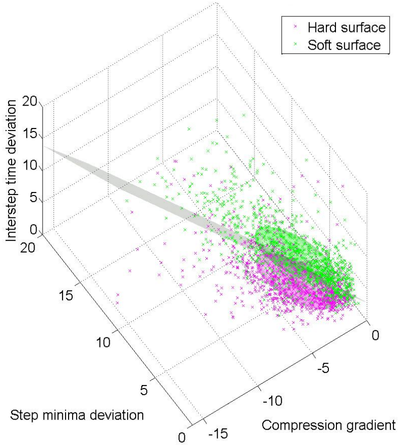
Features 4, 5, 7, 10



Feature 8: mean value per step
Feature 9: median value per step
Feature 11: standard deviation (SD) of the values in one step
Feature 12: SD of the minima
Feature 13: SD of the means
Feature 14: SD of the step standard deviation
Feature 15: SD of the step time
Feature 16: SD of the step area
Feature 17: SD of the time between steps
Feature 18: SD of the time to peak
Feature 19: SD of the time from peak

Classification System: LDA Classifier Visualization

- Visualization of the decision region for hard/soft surface classification in 3D feature space



Shape Modeling

- Each shape is represented by n sampled surface points.
- Surface points are denoted by $\boldsymbol{p}_k \in \mathbb{R}^3$, $k = 1, 2, \ldots, n$.
- The set of surface points is encoded in a single vector (shape vector):

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{p}_1 \\ \boldsymbol{p}_2 \\ \vdots \\ \boldsymbol{p}_n \end{pmatrix} = \begin{pmatrix} p_{1,1} \\ p_{1,2} \\ p_{1,3} \\ p_{2,1} \\ \vdots \\ p_{n,3} \end{pmatrix} \in \mathbb{R}^{3n}$$

with $\boldsymbol{p}_k = (p_{k,1}, p_{k,2}, p_{k,3})^T$.

Shape Modeling (cont.)

We have m shapes, thus m shape vectors, and can generate the landmark configuration matrix:

$$\boldsymbol{L} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m]$$

Shape Modeling (cont.)

We have *m* shapes, thus *m* shape vectors, and can generate the landmark configuration matrix:

$$\boldsymbol{L} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m]$$

Now we can compute the PCA of the columns of $\boldsymbol{L}$ and get the spectral decomposition of the associated covariance matrix

$$\boldsymbol{\Sigma}_L = \sum_i \lambda_i \boldsymbol{e}_i \boldsymbol{e}_i^T$$

where λ_i denote the eigenvalues and $\boldsymbol{e}_i$ the eigenvectors.

Shape Modeling (cont.)

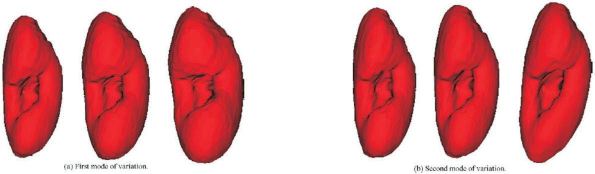
Shape vectors $\boldsymbol{x}^*$ within the eigenvector space can be computed using linear combinations of l eigenvectors:

$$\boldsymbol{x}^* = \bar{\boldsymbol{x}} + \sum_{i=1}^{l} a_i \boldsymbol{e}_i$$

where $\bar{\boldsymbol{x}}$ denotes just the mean of the column vectors of $\boldsymbol{L}$ and $a_i \in \mathbb{R}$ are the shape parameters.

Application of PCA: Segmentation

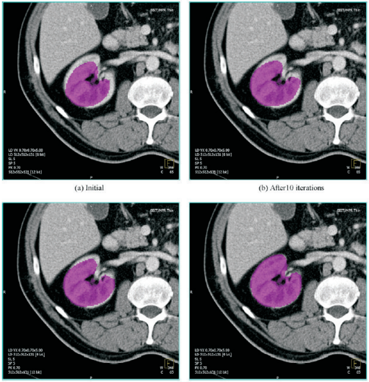
- Lung, liver or kidneys
- Generate and train an Active Shape Model (ASM) for such organs; requires training data and "gold standard" segmentation
- Once point correspondences are found, the different variations within the training data can be easily approximated by its Eigenvectors



M. Spiegel, D. Hahn, V. Daum, J. Wasza, J. Hornegger. "Segmentation of kidneys using a new active shape model generation technique based on non-rigid image registration", Computerized Medical Imaging and Graphics 2009

Fig.: Variation of the mean kidney shape along the first and second Eigenvector.

Application of PCA: Segmentation (cont.)

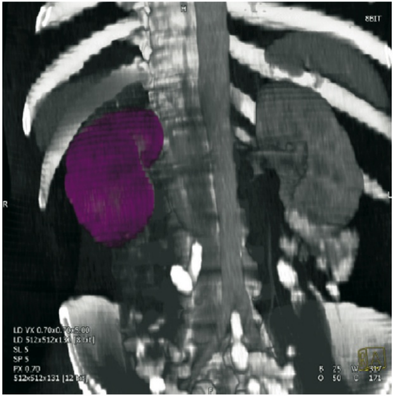


(a) Initial (b) After10 iterations (c) After 20 iterations (d) After 30 iterations and convergence.

M. Spiegel, D. Hahn, V. Daum, J. Wasza, J. Hornegger. “Segmentation of kidneys using a new active shape model generation technique based on non-rigid image registration”, Computerized Medical Imaging and Graphics 2009

Fig.: Iterative segmentation progress of a right kidney using an ASM.

Application of PCA: Segmentation (cont.)



M. Spiegel, D. Hahn, V. Daum, J. Wasza, J. Hornegger. “Segmentation of kidneys using a new active shape model generation technique based on non-rigid image registration”, Computerized Medical Imaging and Graphics 2009

Fig.: 3-D view of the segmentation result.

Pattern Recognition Lab

FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG

TECHNISCHE FAKULTÄT

Next Time in

Pattern Recognition

Notes on Regression

In the two class situation, we set $y \in \{-1, +1\}$ and use the decision rule:

$$y^* = \mathrm{sgn}(\boldsymbol{\alpha}^T \boldsymbol{x} + \alpha_0).$$

We can compute the linear decision boundary simply by least-square estimation.

Linear Regression (cont.)

For a given set of learning data we use matrix notation:

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{x}_1^T & 1 \\ \boldsymbol{x}_2^T & 1 \\ \vdots & \vdots \\ \boldsymbol{x}_m^T & 1 \end{pmatrix} \in \mathbb{R}^{m\times(d+1)} \qquad \boldsymbol{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix} \in \mathbb{R}^m$$

and define

$$\boldsymbol{\theta} = \begin{pmatrix} \boldsymbol{\alpha} \\ \alpha_0 \end{pmatrix}$$

Linear Regression (cont.)

One option to estimate $\boldsymbol{\theta}$ is to solve the linear regression problem:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \|\boldsymbol{X}\boldsymbol{\theta} - \boldsymbol{y}\|_2^2$$

Linear Regression (cont.)

The least-square estimator for the L_2-norm:

$$\begin{aligned}\widehat{\boldsymbol{\theta}} &= \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \sum_{i=1}^{m} (\boldsymbol{\theta}^T \boldsymbol{x}_i - y_i)^2 \\ &= \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \, (\boldsymbol{X}\boldsymbol{\theta} - \boldsymbol{y})^T (\boldsymbol{X}\boldsymbol{\theta} - \boldsymbol{y})\end{aligned}$$

Linear Regression (cont.)

The least-square estimator for the L_2-norm:

$$\begin{aligned}\widehat{\boldsymbol{\theta}} &= \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \sum_{i=1}^{m} (\boldsymbol{\theta}^T \boldsymbol{x}_i - y_i)^2 \\ &= \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \, (\boldsymbol{X}\boldsymbol{\theta} - \boldsymbol{y})^T (\boldsymbol{X}\boldsymbol{\theta} - \boldsymbol{y})\end{aligned}$$

and thus we get

$$\widehat{\boldsymbol{\theta}} = (\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{y}$$

if the column vectors of $\boldsymbol{X}$ are mutually independent.

Linear Regression (cont.)

A few obvious questions:

- Why should we prefer the Euclidean norm (L_2-norm)?
- Will different norms lead to different results?
- Which norm and decision boundary is the best one?
- Can we incorporate prior knowledge in linear regression?

Ridge Regression

In *ridge regression* (also called *regularized regression*) we extend the objective function by an additional term constraining the Euclidean length of the parameter vector $\boldsymbol{\theta}$:

- It is linear regression with the log-likelihood penalized by $-\lambda \boldsymbol{\theta}^T \boldsymbol{\theta}$ where $\lambda > 0$, or alternatively

- It is extended by a prior distribution on the parameter vector $\boldsymbol{\theta}$

$$\boldsymbol{\theta} = \mathcal{N}(0, \mathrm{diag}(\tau^2))$$

Ridge Regression (cont.)

Regularized regression:

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \|\boldsymbol{X\theta} - \boldsymbol{y}\|_2^2 + \lambda \|\boldsymbol{\theta}\|_2^2$$

Ridge Regression (cont.)

Regularized regression:

$$\begin{aligned}\widehat{\boldsymbol{\theta}} &= \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \|\boldsymbol{X}\boldsymbol{\theta} - \boldsymbol{y}\|_2^2 + \lambda \|\boldsymbol{\theta}\|_2^2 \\ &= \underset{\boldsymbol{\theta}}{\operatorname{argmin}} (\boldsymbol{X}\boldsymbol{\theta} - \boldsymbol{y})^T (\boldsymbol{X}\boldsymbol{\theta} - \boldsymbol{y}) + \lambda \cdot \boldsymbol{\theta}^T \boldsymbol{\theta}\end{aligned}$$

Ridge Regression (cont.)

Regularized regression:

$$\begin{aligned}\widehat{\boldsymbol{\theta}} &= \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \|\boldsymbol{X}\boldsymbol{\theta}-\boldsymbol{y}\|_2^2+\lambda\|\boldsymbol{\theta}\|_2^2 \\ &= \underset{\boldsymbol{\theta}}{\operatorname{argmin}} (\boldsymbol{X}\boldsymbol{\theta}-\boldsymbol{y})^T(\boldsymbol{X}\boldsymbol{\theta}-\boldsymbol{y})+\lambda\cdot\boldsymbol{\theta}^T\boldsymbol{\theta}\end{aligned}$$

and thus we get the estimator:

$$\widehat{\boldsymbol{\theta}} = (\boldsymbol{X}^T\boldsymbol{X}+\lambda\boldsymbol{I})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

Ridge Regression (cont.)

Notes:

- The term $\lambda \boldsymbol{I}$ adds a positive constant λ to the diagonal elements.
- The problem is non-singular even if $\boldsymbol{X}^T \boldsymbol{X}$ is not of full rank.
- This was the main motivation of ridge regression when it was first introduced in statistics in 1970.

Ridge Regression (cont.)

Notes:

- The term $\lambda \boldsymbol{I}$ adds a positive constant λ to the diagonal elements.
- The problem is non-singular even if $\boldsymbol{X}^T\boldsymbol{X}$ is not of full rank.
- This was the main motivation of ridge regression when it was first introduced in statistics in 1970.
- The ridge solutions are not equivariant under scaling of the inputs: standardize the input before solving the regression problem!

Ridge Regression (cont.)

Notes:

- The term $\lambda \boldsymbol{I}$ adds a positive constant λ to the diagonal elements.
- The problem is non-singular even if $\boldsymbol{X}^T\boldsymbol{X}$ is not of full rank.
- This was the main motivation of ridge regression when it was first introduced in statistics in 1970.
- The ridge solutions are not equivariant under scaling of the inputs: standardize the input before solving the regression problem!
- The intercept α_0 should not be penalized:
  - Center the input $\boldsymbol{x}_i$.
  - Estimate α_0 by $\bar{y} = \frac{1}{m}\sum_{i=1}^{m} y_i$.
  - Estimate the remaining coefficients by a ridge regression without intercept. Matrix $\boldsymbol{X}$ has d columns (instead of $d+1$).

Ridge Regression (cont.)

Statistical approach: parameters α_j are random variables

- Suppose

$$\forall 1 \leq i \leq m: \quad y_i \sim \mathcal{N}(\underbrace{\boldsymbol{\alpha}^T \boldsymbol{x}_i + \alpha_0}_{\text{mean}}, \underbrace{\sigma^2}_{\text{variance}})$$

- Parameters α_j are assumed to be independent of each other.
- Prior distribution of α_j:

$$\forall 1 \leq j \leq d: \quad \alpha_j \sim \mathcal{N}(0, \tau^2)$$

Ridge Regression (cont.)

Statistical approach: parameters α_j are random variables

- Suppose

$$\forall 1 \leq i \leq m: \quad y_i \sim \mathcal{N}(\underbrace{\boldsymbol{\alpha}^T \boldsymbol{x}_i + \alpha_0}_{\text{mean}}, \underbrace{\sigma^2}_{\text{variance}}) = \frac{1}{\sqrt{2\pi} \cdot \sigma} e^{-\frac{1}{2} \cdot \frac{(y_i - \boldsymbol{\alpha}^T \boldsymbol{x}_i - \alpha_0)^2}{\sigma^2}}$$

- Parameters α_j are assumed to be independent of each other.
- Prior distribution of α_j:

$$\forall 1 \leq j \leq d: \quad \alpha_j \sim \mathcal{N}(0, \tau^2)$$

Ridge Regression (cont.)

Statistical approach: parameters α_j are random variables

- Suppose

$$\forall 1 \leq i \leq m: \quad y_i \sim \mathcal{N}(\underbrace{\boldsymbol{\alpha}^T \boldsymbol{x}_i + \alpha_0}_{\text{mean}}, \underbrace{\sigma^2}_{\text{variance}}) = \frac{1}{\sqrt{2\pi} \cdot \sigma} e^{-\frac{1}{2} \cdot \frac{(y_i - \boldsymbol{\alpha}^T \boldsymbol{x}_i - \alpha_0)^2}{\sigma^2}}$$

- Parameters α_j are assumed to be independent of each other.
- Prior distribution of α_j:

$$\forall 1 \leq j \leq d: \quad \alpha_j \sim \mathcal{N}(0, \tau^2) = \frac{1}{\sqrt{2\pi} \cdot \tau} e^{-\frac{1}{2} \cdot \frac{(\alpha_j - 0)^2}{\tau^2}}$$

Ridge Regression (cont.)

- Maximizing the posterior probability of $\boldsymbol{\alpha}$ for given σ^2 and τ^2:

$$\underset{\boldsymbol{\alpha}}{\operatorname{argmax}} \prod_{i=1}^{m} p(\boldsymbol{\alpha} | y_i)$$

Ridge Regression (cont.)

- Maximizing the posterior probability of $\boldsymbol{\alpha}$ for given σ^2 and τ^2:

$$\underset{\boldsymbol{\alpha}}{\operatorname{argmax}} \prod_{i=1}^{m} p(\boldsymbol{\alpha}|y_i) = \underset{\boldsymbol{\alpha}}{\operatorname{argmax}} \left\{ \prod_{i=1}^{m} p(\boldsymbol{\alpha}) \cdot p(y_i|\boldsymbol{\alpha}) \right\}$$

Ridge Regression (cont.)

- Maximizing the posterior probability of $\boldsymbol{\alpha}$ for given σ^2 and τ^2:

$$
\begin{aligned}
\underset{\boldsymbol{\alpha}}{\operatorname{argmax}} \prod_{i=1}^{m} p(\boldsymbol{\alpha}|y_i) &= \underset{\boldsymbol{\alpha}}{\operatorname{argmax}} \left\{ \prod_{i=1}^{m} p(\boldsymbol{\alpha}) \cdot p(y_i|\boldsymbol{\alpha}) \right\} \\
&= \underset{\boldsymbol{\alpha}}{\operatorname{argmax}} \left\{ \prod_{j=1}^{d} p(\alpha_j) \cdot \prod_{i=1}^{m} p(y_i|\boldsymbol{\alpha}) \right\}
\end{aligned}
$$

Ridge Regression (cont.)

- Maximizing the posterior probability of $\boldsymbol{\alpha}$ for given σ^2 and τ^2:

$$
\begin{aligned}
\underset{\boldsymbol{\alpha}}{\operatorname{argmax}} \prod_{i=1}^{m} p(\boldsymbol{\alpha}|y_i) &= \underset{\boldsymbol{\alpha}}{\operatorname{argmax}} \left\{ \prod_{i=1}^{m} p(\boldsymbol{\alpha}) \cdot p(y_i|\boldsymbol{\alpha}) \right\} \\
&= \underset{\boldsymbol{\alpha}}{\operatorname{argmax}} \left\{ \prod_{j=1}^{d} p(\alpha_j) \cdot \prod_{i=1}^{m} p(y_i|\boldsymbol{\alpha}) \right\} \\
&= \underset{\boldsymbol{\alpha}}{\operatorname{argmax}} \left\{ \sum_{j=1}^{d} \log p(\alpha_j) + \sum_{i=1}^{m} \log p(y_i|\boldsymbol{\alpha}) \right\}
\end{aligned}
$$

Ridge Regression (cont.)

- Maximizing the posterior probability of $\boldsymbol{\alpha}$ for given σ^2 and τ^2:

$$
\begin{aligned}
\underset{\boldsymbol{\alpha}}{\operatorname{argmax}} \prod_{i=1}^{m} p(\boldsymbol{\alpha}|y_i) &= \underset{\boldsymbol{\alpha}}{\operatorname{argmax}} \left\{ \prod_{i=1}^{m} p(\boldsymbol{\alpha}) \cdot p(y_i|\boldsymbol{\alpha}) \right\} \\
&= \underset{\boldsymbol{\alpha}}{\operatorname{argmax}} \left\{ \prod_{j=1}^{d} p(\alpha_j) \cdot \prod_{i=1}^{m} p(y_i|\boldsymbol{\alpha}) \right\} \\
&= \underset{\boldsymbol{\alpha}}{\operatorname{argmax}} \left\{ \sum_{j=1}^{d} \log p(\alpha_j) + \sum_{i=1}^{m} \log p(y_i|\boldsymbol{\alpha}) \right\} \\
&= \underset{\boldsymbol{\alpha}}{\operatorname{argmax}} \left\{ -\frac{1}{2\tau^2} \sum_{j=1}^{d} \alpha_j^2 - \frac{1}{2\sigma^2} \sum_{i=1}^{m} (y_i - \boldsymbol{\alpha}^T \boldsymbol{x} - \alpha_0)^2 \right\}
\end{aligned}
$$

Ridge Regression (cont.)

- Maximizing the posterior probability of $\boldsymbol{\alpha}$ for given σ^2 and τ^2:

$$
\begin{aligned}
\underset{\boldsymbol{\alpha}}{\operatorname{argmax}} \prod_{i=1}^{m} p(\boldsymbol{\alpha}|y_i) &= \underset{\boldsymbol{\alpha}}{\operatorname{argmax}} \left\{ \prod_{i=1}^{m} p(\boldsymbol{\alpha}) \cdot p(y_i|\boldsymbol{\alpha}) \right\} \\
&= \underset{\boldsymbol{\alpha}}{\operatorname{argmax}} \left\{ \prod_{j=1}^{d} p(\alpha_j) \cdot \prod_{i=1}^{m} p(y_i|\boldsymbol{\alpha}) \right\} \\
&= \underset{\boldsymbol{\alpha}}{\operatorname{argmax}} \left\{ \sum_{j=1}^{d} \log p(\alpha_j) + \sum_{i=1}^{m} \log p(y_i|\boldsymbol{\alpha}) \right\} \\
&= \underset{\boldsymbol{\alpha}}{\operatorname{argmax}} \left\{ -\frac{1}{2\tau^2} \sum_{j=1}^{d} \alpha_j^2 - \frac{1}{2\sigma^2} \sum_{i=1}^{m} (y_i - \boldsymbol{\alpha}^T \boldsymbol{x} - \alpha_0)^2 \right\} \\
&= \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \left\{ \frac{\sigma^2}{\tau^2} \sum_{j=1}^{d} \alpha_j^2 + \sum_{i=1}^{m} (y_i - \boldsymbol{\alpha}^T \boldsymbol{x} - \alpha_0)^2 \right\}
\end{aligned}
$$

Ridge Regression (cont.)

- Maximizing the posterior probability of $\boldsymbol{\alpha}$ for given σ^2 and τ^2:

$$\underset{\boldsymbol{\alpha}}{\operatorname{argmax}} \prod_{i=1}^{m} p(\boldsymbol{\alpha}|y_i) = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \left\{ \lambda \boldsymbol{\alpha}^T \boldsymbol{\alpha} + (\boldsymbol{X}\boldsymbol{\alpha} - \boldsymbol{y})^T (\boldsymbol{X}\boldsymbol{\alpha} - \boldsymbol{y}) \right\} \quad \text{with} \quad \lambda = \frac{\sigma^2}{\tau^2}$$

Ridge Regression (cont.)

- Maximizing the posterior probability of $\boldsymbol{\alpha}$ for given σ^2 and τ^2:

$$\underset{\boldsymbol{\alpha}}{\operatorname{argmax}} \prod_{i=1}^{m} p(\boldsymbol{\alpha}|y_i) = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \left\{ \lambda \boldsymbol{\alpha}^T \boldsymbol{\alpha} + (\boldsymbol{X}\boldsymbol{\alpha} - \boldsymbol{y})^T (\boldsymbol{X}\boldsymbol{\alpha} - \boldsymbol{y}) \right\} \quad \text{with} \quad \lambda = \frac{\sigma^2}{\tau^2}$$

- The ridge estimate is the mode of the posterior pdf!

Lasso

Regularized regression using a mixture of L_2- and L_1-norm, where the residual is penalized using the L_2-norm and the regularizer uses the L_1-norm:

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \|\boldsymbol{X}\boldsymbol{\theta} - \boldsymbol{y}\|_2^2 + \lambda \cdot \|\boldsymbol{\theta}\|_1$$

Lasso

Regularized regression using a mixture of L_2- and L_1-norm, where the residual is penalized using the L_2-norm and the regularizer uses the L_1-norm:

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \|\boldsymbol{X}\boldsymbol{\theta} - \boldsymbol{y}\|_2^2 + \lambda \cdot \|\boldsymbol{\theta}\|_1$$

The lasso is used to compute a sparse solution of the system of linear equations, i. e. the number of non-zero elements in $\boldsymbol{\theta}$ shall be small.

Lessons Learned

- Principal component analysis
- Linear discriminant analysis with and without dimension reduction
- Both PCA and LDA relate to an eigenvalue eigenvector problem
- Alternative formulation of LDA using the Fisher transform
- Linear and ridge regression for classification

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.

Further Readings (cont.)

Details on the adidas_1 shoe and the implemented classifier:

- B. Eskofier, F. Hönig, P. Kühner:
  Classification of Perceived Running Fatigue in Digital Sports,
  Proceedings of the 19th International Conference on Pattern Recognition (ICPR 2008), Tampa, Florida, U. S. A., 2008

Details on the shape modeling of kidneys and its application to segmentation:

- M. Spiegel, D. Hahn, V. Daum, J. Wasza, J. Hornegger:
  Segmentation of kidneys using a new active shape model generation technique based on non-rigid image registration,
  Computerized Medical Imaging and Graphics 2009 33(1):29-39

Comprehensive Questions

- What is the difference between PCA and LDA?
- How can PCA and LDA be combined to achieve a high rank reduction?
- Write down a straight forward objective function for linear regression!
- What happens if we replace the L_2-norm by another norm?