LINEAR DISCRIMINANCE ANALYSIS

Christoph Bandt 12.10.2011

1. Separating hyperplane
2. Statistical problem
3. Curse of dimension?
4. Tensor product LDA

1. Linear classification problem
   \( n_1 + n_2 \) objects from 2 classes
   Objects given by data
   \( \mathbf{w} \rightarrow \text{identified with vectors} \)
   \( \mathbf{x}^{(ii)} \in \mathbb{R}^p \quad i = 1 \ldots n_1 \text{ class } 1 \)
   \( i = n_1 + 1 \ldots n_1 + n_2 \text{ class } 2 \)

Problem: find a linear function
\( g: \mathbb{R}^p \rightarrow \mathbb{R} \) and a constant \( c \in \mathbb{R} \)
such that
\( g(x^{(ii)}) < c \) for class 1
\( g(x^{(ii)}) > c \) for class 2

\[ g(x^1) = c \]
\[ k_1 = \text{conv} \{ x^{(i)} \mid i = 1 \ldots n_1 \} \]

\[ k_2 = \text{conv} \{ x^{(i)} \mid i = n_1 + 1 \ldots n \} \]

**Th. (Hahn–Banach)**

If \( k_1 \cap k_2 = \emptyset \), there exists a separating \( g(x) = 0 \).

**Rem.** If \( n > p + 1 \), the problem usually cannot be solved because \( k_1 \cap k_2 \neq \emptyset \).

In high dimension, \( p \geq n - 1 \), the sets are in general disjoint.

**Ex.** \( n = 4, \ p = 3 \)

\( n_1 = 3, \ n_2 = 1 \)

\( n_4 = n_2 = 2 \)

Determine the separation by linear programming.
2) Training and test data

$x^{(i)}$ are training data, used to find $g(x_1 = c$ but separation should work for other data $y \in \mathbb{R}^p$.

The $y$ are called test data.

This problem can only be solved when training and test data are similar, or have common properties. Statistical question!

Modeling: Assumptions!

There are two density functions $f_1(x), f_2(x)$ on $\mathbb{R}^p$, and the data from class 1 and 2 are independent random vectors drawn from these density functions. $f_1, f_2$ describe populations of objects.
Solution for the two populations
(Neyman + Pearson & 1938)
The best \( g \) is given as
\[
g(x) = \frac{f_2(x)}{f_1(x)} \text{ and appropriate C.}
\]

Discriminance analysis
(Fisher + Welsh & 1940)
When \( f_i \) are normal distribution, then \( g(x) \) is quadratic function.
When the covariance matrix is equal, then \( g(x) \) is linear.

Fisher shows, that for normal distributions with equal covariance the best possible classification is by a linear function.

This is theory!
For the data we need to estimate \( f_1, f_2 \).
For normal distribution: \( p_i = \Sigma_i \).
For $i$, the estimate is

$$\hat{\mu}_i = \bar{x}^{(i)}$$

The sample mean of training data

Real problem is estimate of $\Sigma$

We assume $\Sigma_1 = \Sigma_2$

Then estimate is pooled sample covariance $w_X$

$$S = \frac{1}{n-2} D' w_X D$$

These is the centred data $w_X$

$$D = \begin{pmatrix}
X^{(1)} - \bar{x}^{(1)} \\
X^{(2)} - \bar{x}^{(2)} \\
\vdots \\
X^{(n)} - \bar{x}^{(n)} \\
\end{pmatrix}
$$

Result:

$$g(x) = S^{-1} \left( \bar{x} - \bar{x}^{(2)} \right)$$

$$c = g \left( \frac{\bar{x}^{(1)}}{2} + \bar{x}^{(2)} \right)$$
3) high dimension $p \gg n$

For example, 2x50 object in R^1000

known: Fisher LDA does not work

Reason: A small sample cannot contain all the information to estimate $\Sigma$. Given: \( n \times p \) numbers

To estimate: for $\mu_1, \mu_2$ $2p$ values

for $\Sigma$: \( \frac{p(p+1)}{2} \) covariances too much

In $\mathbb{R}^P$, the training data are on one co-dimensional plane. No information on variance perpendicular to $\Pi$. !
**Revis. Problem is to estimate \( \Sigma \)**

There is no curse of dimension because separation is easy.

**Solution a)** Regularized LDA

\[ \Sigma = (1- \lambda) S + \lambda I \]

**Assumption:**

Variance \( \lambda \) in every direction

Best \( \lambda \) by optimization

(Stefan)

---

1) Further assumptions on structure of \( \Sigma \)

For example, diagonal \( \Sigma = \begin{pmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{pmatrix} \)

only \( p \) parameters

as block matrix

\[ \Sigma = \begin{pmatrix} \Sigma_{p_1} & 0 \\ 0 & \Sigma_{p_m} \end{pmatrix} \]

where \( \Sigma_{p_i} \) is \( p_i \times p_i \)

Example: \( p_i = \sqrt{p} = m : \sqrt{p} \cdot p = \frac{3}{2} \) parameters
But: it is better to assume equations between entries of $\Sigma$
(symmetric of $\Sigma$).
Equations allows for more accurate estimates.

Reen, (Combinatorial problem)
Given a $0-1-n\times l$ matrix (1 sign at $i,j$)
find a reordering of rows and columns
such that the result is (almost) a block matrix.

4) Tensor product LDA
Assume
\[ \Sigma = U \otimes V \]

\[ U = \begin{pmatrix} u_{11} & \cdots & u_{1n} \\ \vdots & \ddots & \vdots \\ u_{m1} & \cdots & u_{mn} \end{pmatrix} \text{ and } V \text{ qrankless matrix} \]

\[ U \otimes V = \begin{pmatrix} u_{11} V & u_{12} V & \cdots & u_{1n} V \\ \vdots & \vdots & \ddots & \vdots \\ u_{m1} V & u_{m2} V & \cdots & u_{mn} V \end{pmatrix} \]
EEG:
- Spatial structure (32 channels)
- Temporal structure (by stationarity, further equal)

Data problem: Do our EEG data really have tensor product structure? If not, where are the differences?

Prop: In case of product structure, Hey's method gives the Fisher LDA (two-step LDA)

Advantage of product
- fewer parameters
  \( R_p = \text{vec}(U) \otimes \text{vec}(V) \)
  \( 2 \binom{p}{2} \times r \) parameters
- Parameters can be estimated from pooled data
\[ \sum^{-1} = U^{-1} \otimes V^{-1} \]

\[
d = \hat{\mu}_2 - \hat{\mu}_1 = (d_1, d_2, \ldots, d_T)'
\]

\[
d_t = (d_{t1}, \ldots, d_{tN})
\]

\[
\hat{\omega} = \sum^{-1} d = \left( \frac{1}{\sum w_{jt}^t} V_j^t \right)_{j=1}^{N_t}
\]

\[
\hat{\omega}_t = V^t d_t
\]

1st step CDA for every \( t \)

2nd step CDA with the scores \( \hat{\omega}_t \)