Cognitive Algorithms
Lecture 3

Linear Classification

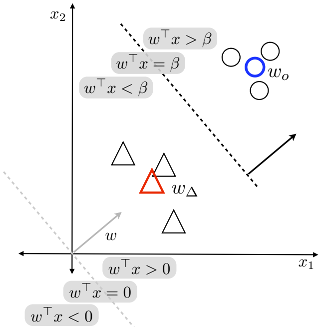
Klaus-Robert Müller, Wojciech Samek
Stephanie Brandl

Berlin Institute of Technology
Dept. Machine Learning

Summary Lecture 2

- Biological Neural Networks
  - Cascade of (non-linear) filters of sensory features
  - Abstract ideas are based on integration of these features
  - How integration is done is subject of neuroscientific research
- Psychologists postulated we learn **Prototypes**
  - Prototypes can be the class means
  - New data is associated with **closest** Prototype
  - Prototype theory is closely related to linear classification
- Artificial Neural Networks
  - Inspired by biological neural networks
  - Perceptron algorithm realizes linear classification

Linear Classification Revisited



Comparison of distance to class means is equivalent to linear classification

$$\|\mathbf{x} - \mathbf{w}_\Delta\| > \|\mathbf{x} - \mathbf{w}_o\|$$
$$\Leftrightarrow 0 < \mathbf{w}^\top \mathbf{x} - \beta$$

where

$$\mathbf{w} = \mathbf{w}_o - \mathbf{w}_\Delta$$

and

$$\beta = 1/2 \cdot (\mathbf{w}_o^\top \mathbf{w}_o - \mathbf{w}_\Delta^\top \mathbf{w}_\Delta)$$
$$= 1/2 \cdot \mathbf{w}^\top (\mathbf{w}_o + \mathbf{w}_\Delta)$$

This simple linear classification rule is often called **Nearest Centroid Classifier**.

Perceptron Algorithm with Stochastic Gradient Descent

Computes: Normal vector $\mathbf{w}$ of decision hyperplane for binary classification

Input: Data $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}, \mathbf{x}_i \in \mathbb{R}^D,\ y_i \in \{-1, +1\}$,
learning rate η,
iterations N_{it}

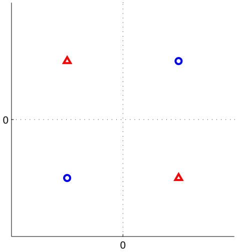
Algorithm:

$\mathbf{w} = \mathbb{1}/D$
for $i = 1$ **to** N_{it} **do**
  Pick a random data point $\mathbf{x}_i$
  if $\mathbf{w}^\top \mathbf{x}_i \cdot y_i < 0$ **then**
    $\mathbf{w} = \mathbf{w} + \eta / i \cdot \mathbf{x}_i \cdot \mathbf{y}_i$
  end if
end for

Output: $\mathbf{w}$

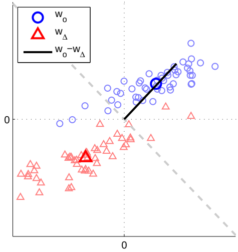
Problems with Nearest Centroid Classification

Non-linear Data



Solutions
Non-linear features,
Non-linear classification methods

Correlated Data



Solution
(Fisher's) Linear Discriminant Analysis

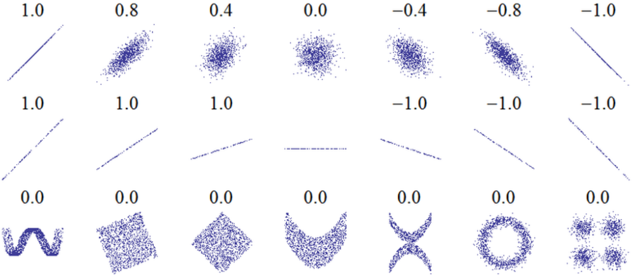
Covariance and Correlation

For two random variables X and Y, their **covariance** and **correlation** are defined as

$$\mathrm{Cov}(X, Y) := \mathrm{E}[(X - \mathrm{E}(X))(Y - \mathrm{E}(Y))]$$

$$\mathrm{Corr}(X, Y) := \frac{\mathrm{Cov}(X, Y)}{\sqrt{\mathrm{Var}(X)}\sqrt{\mathrm{Var}(Y)}}.$$

Correlation measures the linear relationship between X and Y:

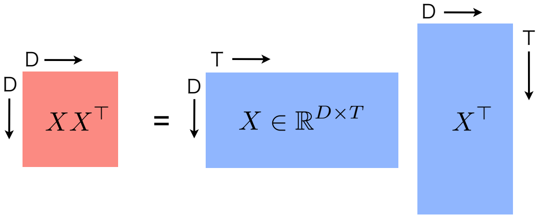


Covariance Matrices

Given T data points $\mathbf{x}_t \in \mathbb{R}^D$ in a data matrix $X \in \mathbb{R}^{D\times T}$ the empirical estimate of the **covariance matrix** is defined as

$$S = \frac{1}{T} \; XX^\top \tag{1}$$

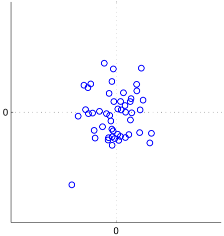
where we assume centered data, i.e. $\sum_{t=1}^{T} \mathbf{x}_t = \mathbf{0}$.



Correlated Data and Linear Mappings

We can generate correlated data using a diagonal scaling matrix D and a rotation R

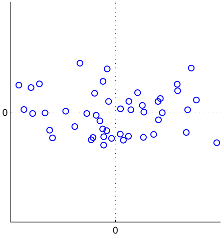
Uncorrelated



$$x \sim \mathcal{N}(0, 1)$$

$$XX^\top = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

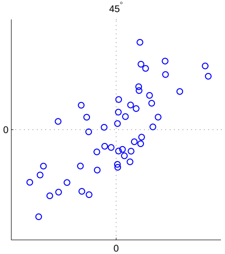
Uncorrelated, scaled



$$\begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix} X$$

$$XX^\top = \begin{bmatrix} 9 & 0 \\ 0 & 1 \end{bmatrix}$$

Scaled, rotated by 45°



$$\begin{bmatrix} cos(\phi) & -sin(\phi) \\ sin(\phi) & cos(\phi) \end{bmatrix} \begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix} X$$

$$XX^\top = \begin{bmatrix} 5 & 4 \\ 4 & 5 \end{bmatrix}$$

Ronald A. Fisher

R.A. Fisher (1890 - 1962)

Founder of modern statistics

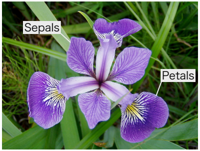
Interested in Biology

Suggested *Linear Discriminant Analysis* (LDA) [Fisher, 1936]

The *Iris* Flower Dataset

Iris Setosa

Iris Versicolor

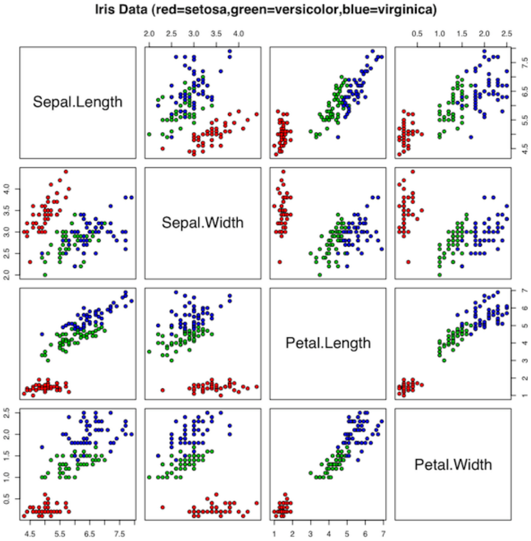


Iris Virginica

`http://en.wikipedia.org/wiki/Iris_flower_data_set`

- 50 flowers of each species were collected
  "all from the same pasture, and picked on the same day and measured at the same time by the same person with the same apparatus"
- Petal and Sepal length and width were measured
- Very popular benchmark data set

The *Iris* Flower Dataset

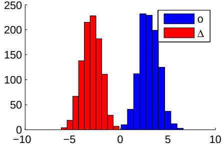


http://en.wikipedia.org/wiki/Iris_flower_data_set

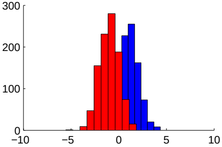
The Fisher Criterion - measure for class separability

Consider one dimensional data and two classes

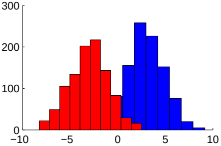
Good Class Separation



Bad Class Separation: Close means



Bad Class Separation: Large Variance



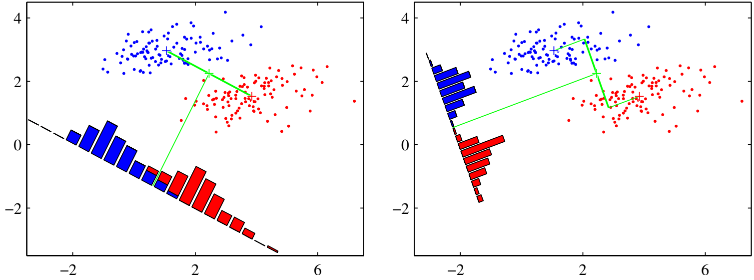
The fisher criterion:

$$\frac{\text{between class variance}}{\text{within class variance}} = \frac{(\mathbf{w}_o - \mathbf{w}_\Delta)^2}{\sigma_o^2 + \sigma_\Delta^2}$$

where $\mathbf{x}_{1o}, \ldots, \mathbf{x}_{N_o o} \in \mathbb{R}^D$ and
$\mathbf{w}_o = \frac{1}{N_o} \sum_{i=1}^{N_o} \mathbf{x}_{io}$ and $\sigma_o^2 = \frac{1}{N_o} \sum_{i=1}^{N_o} (\mathbf{x}_{io} - \mathbf{w}_o)^2$.

Linear Discriminant Analysis

View classification in terms of dimensionality reduction



Goal: Find a (normal vector of a linear decision boundary) $\mathbf{w} \in \mathbb{R}^D$ that

Maximizes mean class difference, and

Minimizes variance in each class

Linear Discriminant Analysis

Goal: Find a (normal vector of a linear decision boundary) $\mathbf{w} \in \mathbb{R}^D$ that

Maximizes mean class difference

$$(\mathbf{w}^\top \mathbf{w}_o - \mathbf{w}^\top \mathbf{w}_\Delta)^2 = \mathbf{w}^\top \underbrace{(\mathbf{w}_o - \mathbf{w}_\Delta)(\mathbf{w}_o - \mathbf{w}_\Delta)^\top}_{S_B - \text{"between class scatter"}} \mathbf{w} \tag{2}$$

Minimizes variance in each class

$$\frac{1}{N_o} \sum_{i=1}^{N_o} \left(\mathbf{w}^\top (\mathbf{x}_{oi} - \mathbf{w}_o) \right)^2 + \frac{1}{N_\Delta} \sum_{j=1}^{N_\Delta} \left(\mathbf{w}^\top (\mathbf{x}_{\Delta j} - \mathbf{w}_\Delta) \right)^2$$

$$= \mathbf{w}^\top \underbrace{\left(\frac{1}{N_o} \sum_{i=1}^{N_o} (\mathbf{x}_{oi} - \mathbf{w}_o)(\mathbf{x}_{oi} - \mathbf{w}_o)^\top + \frac{1}{N_\Delta} \sum_{j=1}^{N_\Delta} (\mathbf{x}_{\Delta j} - \mathbf{w}_\Delta)(\mathbf{x}_{\Delta j} - \mathbf{w}_\Delta)^\top \right)}_{S_W - \text{"within class scatter"}} \mathbf{w}$$

Linear Discriminant Analysis

Goal: Find a (normal vector of a linear decision boundary) $\mathbf{w}$ that

Maximizes mean class difference, $\mathbf{w}^\top S_B \mathbf{w}$ and

Minimizes variance in each class, $\mathbf{w}^\top S_W \mathbf{w}$

$\rightarrow$ maximize the *Fisher criterion*

$$\underset{\mathbf{w}}{\operatorname{argmax}} \frac{\mathbf{w}^\top S_B \mathbf{w}}{\mathbf{w}^\top S_W \mathbf{w}} \tag{3}$$

Linear Discriminant Analysis

$$\underset{\mathbf{w}}{\operatorname{argmax}} \frac{\mathbf{w}^\top S_B \mathbf{w}}{\mathbf{w}^\top S_W \mathbf{w}}$$

To optimize the Fisher criterion, we set its derivative w.r.t $\mathbf{w}$ to 0

$$\begin{aligned}
\frac{(\mathbf{w}^\top S_W \mathbf{w}) S_B \mathbf{w} - (\mathbf{w}^\top S_B \mathbf{w}) S_W \mathbf{w}}{(\mathbf{w}^\top S_W \mathbf{w})^2} &= 0 \\
(\mathbf{w}^\top S_B \mathbf{w}) S_W \mathbf{w} &= (\mathbf{w}^\top S_W \mathbf{w}) S_B \mathbf{w} \\
S_W \mathbf{w} &= S_B \mathbf{w} \underbrace{\frac{\mathbf{w}^\top S_W \mathbf{w}}{\mathbf{w}^\top S_B \mathbf{w}}}_{scalar}
\end{aligned}$$

Linear Discriminant Analysis

$$\underset{\mathbf{w}}{\operatorname{argmax}} \frac{\mathbf{w}^\top S_B \mathbf{w}}{\mathbf{w}^\top S_W \mathbf{w}}$$
$$\rightarrow S_W \mathbf{w} = S_B \mathbf{w} \lambda$$

Note that

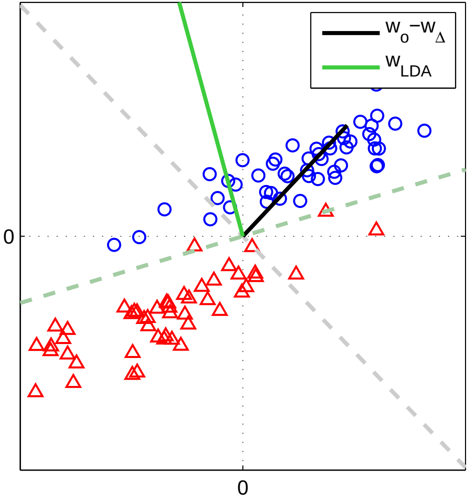
$$S_B \mathbf{w} = (\mathbf{w}_o - \mathbf{w}_\Delta) \underbrace{(\mathbf{w}_o - \mathbf{w}_\Delta)^\top \mathbf{w}}_{\text{scalar}}$$

thus left multiplying with S_W^{-1} yields

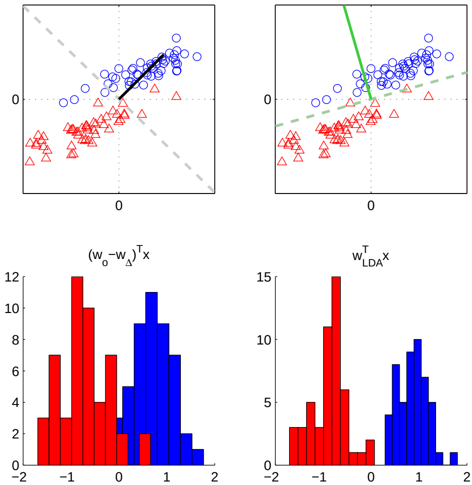
$$\mathbf{w} \propto S_W^{-1} (\mathbf{w}_o - \mathbf{w}_\Delta).$$

($\propto$ denotes proportional)

Linear Discriminant Analysis vs Nearest Centroid Classifier



Linear Discriminant Analysis vs Nearest Centroid Classifier



Linear Discriminant Analysis

If both classes have the same covariance matrix, LDA first *decorrelates* the data followed by nearest centroid classification:

$$\begin{aligned} \mathbf{x} &\mapsto \operatorname{sign}(\mathbf{w}^T \cdot \mathbf{x} - \beta) \\ \mathbf{w} &\propto S_W^{-1}(\mathbf{w}_o - \mathbf{w}_\Delta) \end{aligned}$$

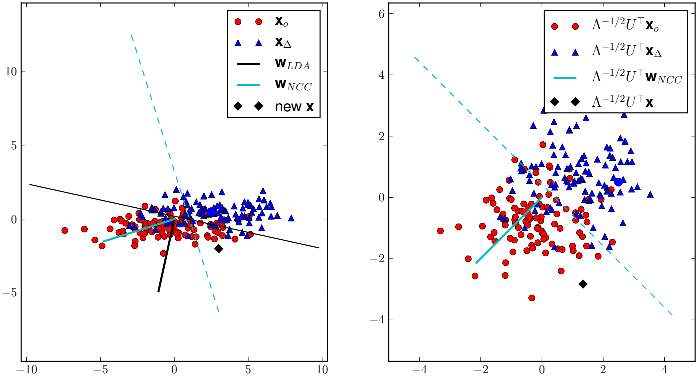
$$\mathbf{w}^T\mathbf{x} = (\mathbf{w}_o - \mathbf{w}_\Delta)^T S_W^{-1}\mathbf{x} = \underbrace{(\mathbf{w}_o - \mathbf{w}_\Delta)^T U\Lambda^{-1/2}}_{\substack{\text{mean class difference} \\ \text{of decorrelated data}}} \underbrace{\Lambda^{-1/2}U^T\mathbf{x}}_{\text{decorrelated } \mathbf{x}}$$

where $S_W = U\Lambda U^T$ is the eigenvalue decompostion of S_W

Linear Discriminant Analysis

LDA first *decorrelates* the data followed by nearest centroid classification:

$$\mathbf{w}^T\mathbf{x} = (\mathbf{w}_o - \mathbf{w}_\Delta)^T S_W^{-1}\mathbf{x} = \underbrace{(\mathbf{w}_o - \mathbf{w}_\Delta)^T U\Lambda^{-1/2}}_{\text{mean class difference of decorrelated data}} \underbrace{\Lambda^{-1/2}U^T\mathbf{x}}_{\text{decorrelated } \mathbf{x}}$$



Decision theory

Decision theory:
For a new data point $\mathbf{x} \in \mathbb{R}^D$

$$\text{Decide class } \Delta \qquad \text{if } p(\Delta|\mathbf{x}) > p(o|\mathbf{x}).$$

Calculate $p(\Delta|\mathbf{x})$ with Bayes rule:

$$\begin{aligned} p(\Delta|\mathbf{x}) &= \frac{p(\Delta, \mathbf{x})}{p(\mathbf{x})} \\ &= \frac{p(\Delta)p(\mathbf{x}|\Delta)}{p(\mathbf{x})} \end{aligned}$$

Decision theory

Estimating $p(\mathbf{x}|\Delta)$ is difficult: already if each dimension of $\mathbf{x}$ can take 2 values $\rightarrow 2^D$ possible values.

One possibilty to deal with it:

Choose a distribution $p(\mathbf{x}|\Delta)$, $p(\mathbf{x}|o)$ that is easy to deal with

$\rightarrow$ Most popular: The Gaussian (or Normal) distribution

$$\mathbf{x} \in \mathbb{R}^D \sim \mathcal{N}(\mathbf{w}_\Delta, S_\Delta) = \frac{1}{(2\pi)^{\frac{D}{2}} \sqrt{|S_\Delta|}} e^{-\frac{1}{2}(\mathbf{x}-\mathbf{w}_\Delta)^\top S_\Delta^{-1}(\mathbf{x}-\mathbf{w}_\Delta)}$$

Linear Discriminant - a Probabilistic View

If we assume equal covariance in each class, $S_W = 2S_\Delta = 2S_o$, and equal class probablities, $p(\Delta) = p(o) = 0.5$, the optimal classification boundary is linear and given by

$$\begin{aligned}
\mathbf{w} &= S_W^{-1}(\mathbf{w}_o - \mathbf{w}_\Delta) \\
\beta &= \frac{1}{2}\mathbf{w}_o S_W^{-1}\mathbf{w}_o - \frac{1}{2}\mathbf{w}_\Delta S_W^{-1}\mathbf{w}_\Delta = \frac{1}{2}\mathbf{w}^T(\mathbf{w}_o + \mathbf{w}_\Delta)
\end{aligned}$$

$\Rightarrow$ Linear decision boundaries arise from simple assumption about the distribution of the data.

Linear Discriminant - a Probabilistic View

If we assume equal covariance in each class, $S_W = 2S_\Delta = 2S_o$, the optimal classification boundary is linear and given by

$$\begin{aligned}
\mathbf{w} &= S_W^{-1}(\mathbf{w}_o - \mathbf{w}_\Delta) \\
\beta &= \frac{1}{2}\mathbf{w}_o S_W^{-1}\mathbf{w}_o - \frac{1}{2}\mathbf{w}_\Delta S_W^{-1}\mathbf{w}_\Delta + \log\frac{p(\Delta)}{p(o)} \\
&= \frac{1}{2}\mathbf{w}^T(\mathbf{w}_o + \mathbf{w}_\Delta) + \log\frac{p(\Delta)}{p(o)}
\end{aligned}$$

$\Rightarrow$ Linear decision boundaries arise from simple assumption about the distribution of the data.

Linear Discriminant Algorithm

Computes: Normal vector $\mathbf{w}$ of decision hyperplane, threshold β

Input: Data $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}, x_i \in \mathbb{R}^D,\ y_i \in \{-1, +1\}$,

Compute class mean vectors

$\mathbf{w}_{-1} = 1/N_- \sum_{i \in \mathcal{Y}_{-1}} \mathbf{x}_i$

$\mathbf{w}_{+1} = 1/N_+ \sum_{j \in \mathcal{Y}_{+1}} \mathbf{x}_j$

Compute *within-class* covariance matrices

$S_W = 1/N_- \sum_{i \in \mathcal{Y}_{-1}} (\mathbf{x}_i - \mathbf{w}_{-1})(\mathbf{x}_i - \mathbf{w}_{-1})^\top$

$\quad +1/N_+ \sum_{j \in \mathcal{Y}_{+1}} (\mathbf{x}_j - \mathbf{w}_{+1})(\mathbf{x}_j - \mathbf{w}_{+1})^\top$

Compute normal vector $\mathbf{w}$

$\mathbf{w} = S_W^{-1}(\mathbf{w}_{+1} - \mathbf{w}_{-1})$

Compute threshold

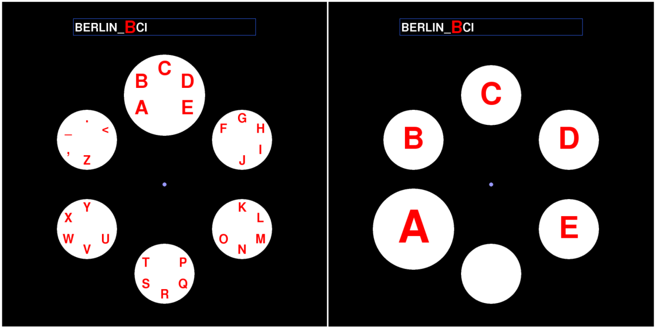
$\beta = 1/2\ \mathbf{w}^T (w_{+1} + \mathbf{w}_{-1}) + \log(N_-/N_+)$

Output: $\mathbf{w}$, β

Berlin Brain-Computer-Interface (BBCI)

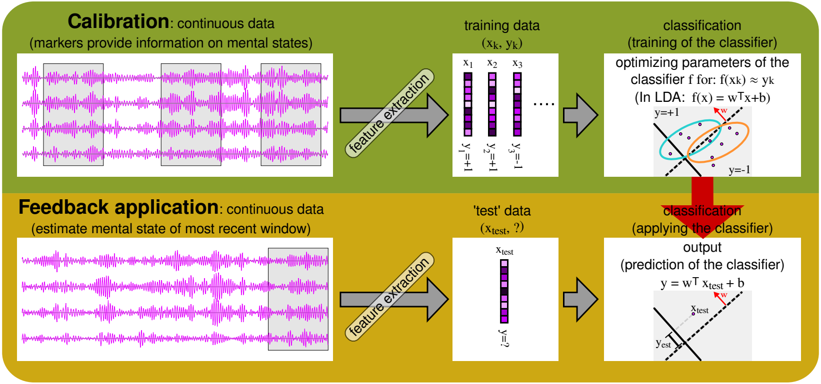
Hex-o-spell: Writing with thoughts

`http://www.bbci.de/`



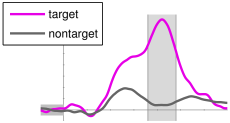
Demo: `http://iopscience.iop.org/1741-2552/8/6/066003/media`

BCI with ML: Calibration and Feedback



BCI Based on Event-Related Potentials (ERPs)

- User concentrates on a symbol
- Rows and columns are intensified randomly
- Target rows and columns elicit specific ERPs
- BCI detects target ERPs (averaged over few repetitions)



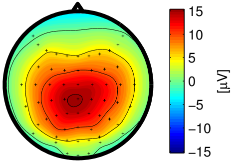
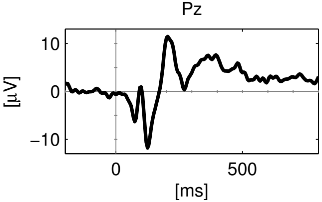
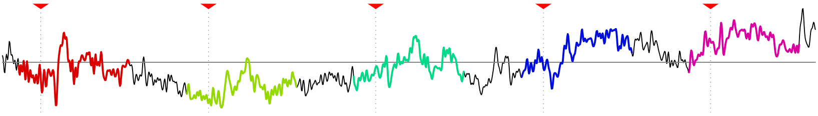


Illustration: Single-Trials and ERPs



Continous Signal (with markers):



Segments (epochs) around stimulus markers:

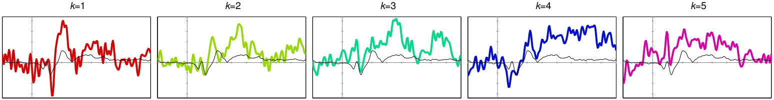
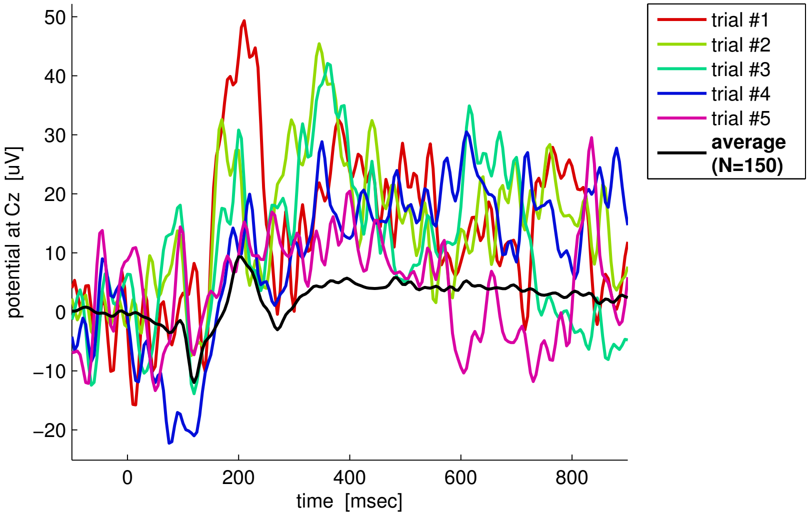
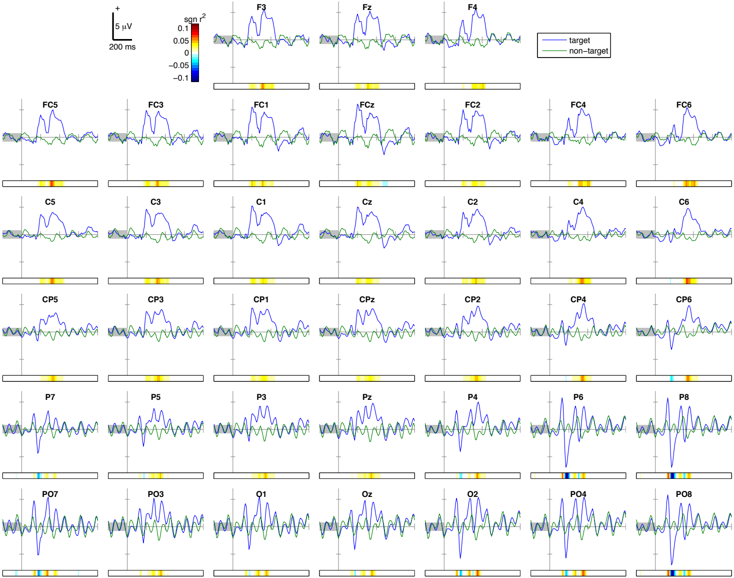


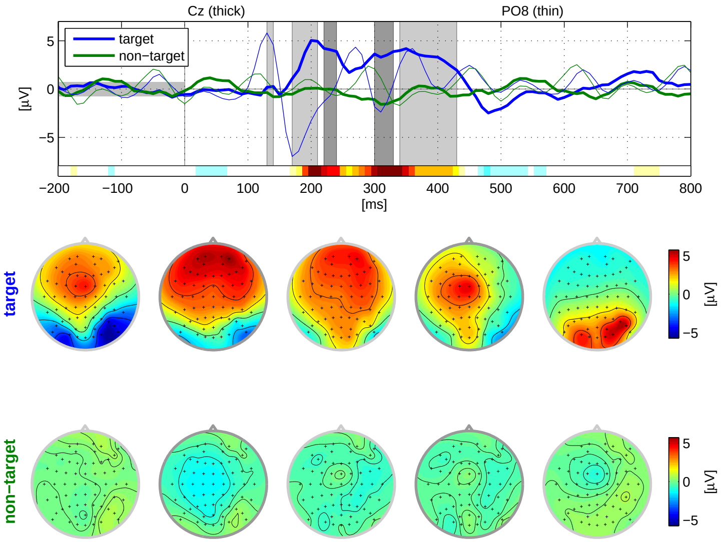
Illustration: Single-Trials and ERPs



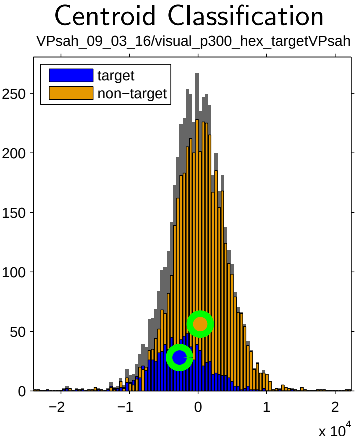
Scalp Potentials In Response to Targets/Non-Targets

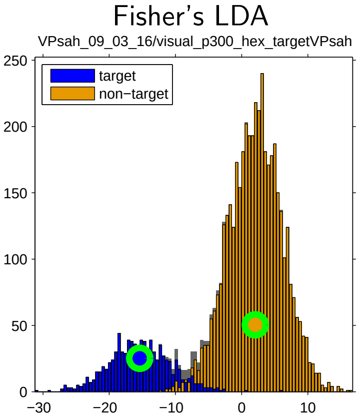


Berlin Brain-Computer-Interface

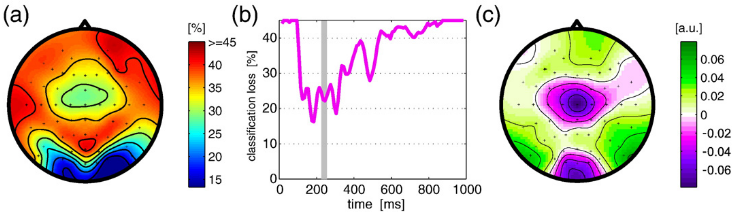


Berlin Brain-Computer-Interface





Understanding the classifier



(a) Classification error on features from the time interval 115-535m
(b) Classification error for intervals of 30ms duration
(c) Weight vector of classification on features from the time interval 220-250ms
[Blankertz et al., 2011]

Generalization and Model Evaluation

The goal of classification is **generalization**: Correct categorization/prediction of new data

How can we estimate generalization performance?

$\rightarrow$ **Cross-validation**:

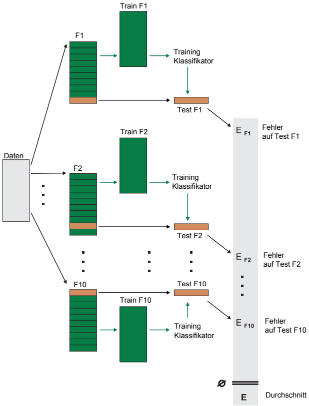
- Train model on part of data
- Test model on other part of data
- Repeat on different cross-validation *folds*
- Average performance on test set across all folds

Cross-Validation

Algorithm 1: Cross-Validation

Require: Data $(x_1, y_1) \ldots, (x_N, y_N)$, Number of CV folds F

1: # Split data in F **disjunct** folds
2: **for** folds $f = 1, \ldots, F$ **do**
3: # Train model on folds $\{1, \ldots, F\} \setminus f$
4: # Compute prediction error on fold f
5: **end for**
6: # Average prediction error



Summary

Correlations between features can affect classification accuracy

Fisher proposed Linear Discriminant Analysis (LDA)

LDA maximizes *between class variance* while minimizing *within class variance*

If data is Gaussian with equal class covariances, than LDA is the optimal classifer

LDA is used in state-of-the-art BCI systems

We can use Cross-validation for Model Evaluation

References

B. Blankertz, S. Lemm, M. Treder, S. Haufe, and K.-R. Müller. Single-trial analysis and classification of erp components–a tutorial. *Neuroimage*, 56(2):814–25, 2011. doi: 10.1016/j.neuroimage.2010.06.048.

R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936.