Machine Learning

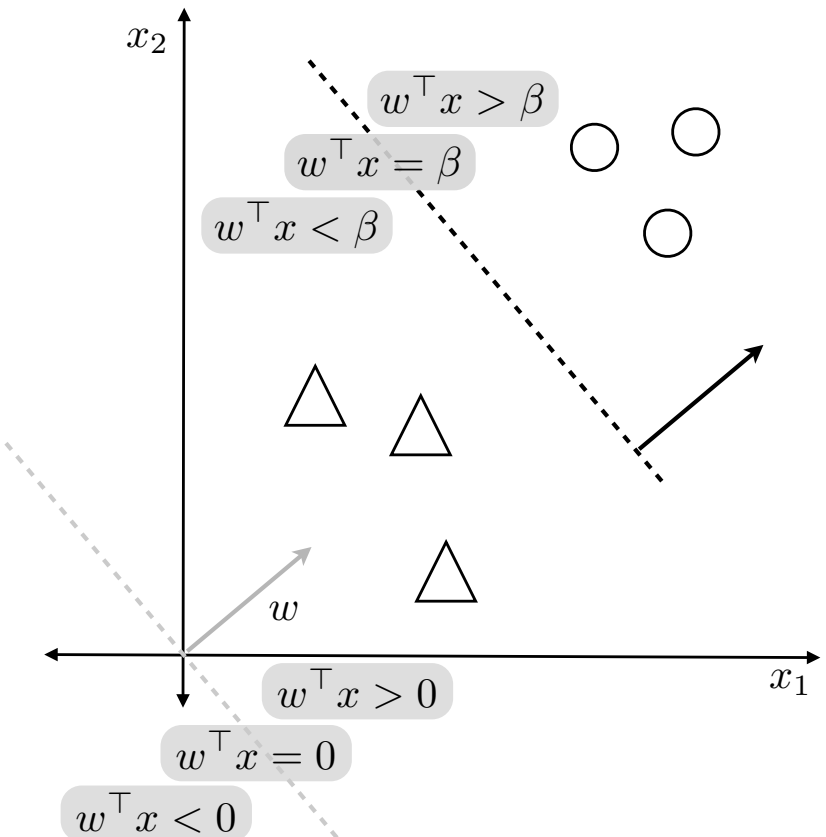
Lecture 7
Linear Discriminant Analysis

Felix Bießmann

Beuth University & Einstein Center for Digital Future

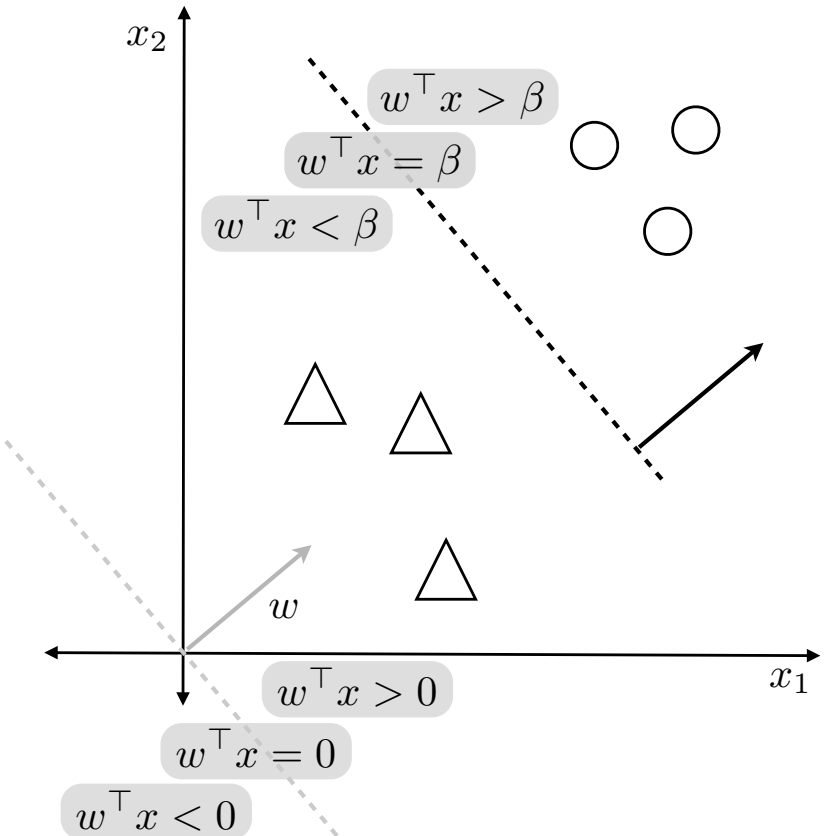
May 26, 2019

Linear Classification



$$\mathbf{w}^\top \mathbf{x} - \beta = \begin{cases} > 0 & \text{if } \mathbf{x} \text{ belongs to } o \\ < 0 & \text{if } \mathbf{x} \text{ belongs to } \Delta \end{cases}$$

Linear Classification



$$\mathbf{w}^\top \mathbf{x} - \beta = \begin{cases} > 0 & \text{if } \mathbf{x} \text{ belongs to } o \\ < 0 & \text{if } \mathbf{x} \text{ belongs to } \Delta \end{cases}$$

The *offset* β can be included in $\mathbf{w}$

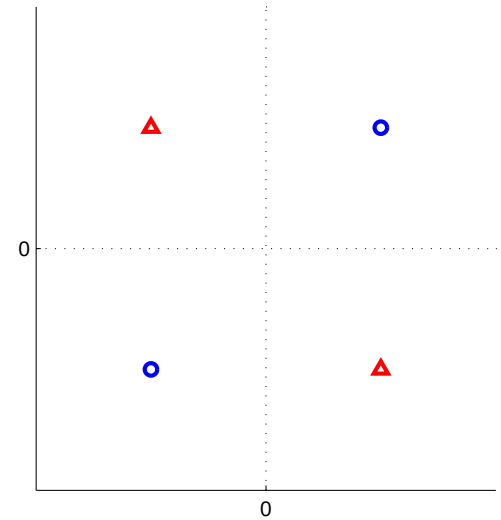
$$\tilde{\mathbf{x}} \leftarrow \begin{bmatrix} 1 \\ \mathbf{x} \end{bmatrix} \qquad \tilde{\mathbf{w}} \leftarrow \begin{bmatrix} -\beta \\ \mathbf{w} \end{bmatrix}$$

such that

$$\tilde{\mathbf{w}}^\top \tilde{\mathbf{x}} = \mathbf{w}^\top \mathbf{x} - \beta.$$

Problems with Linear Classifiers

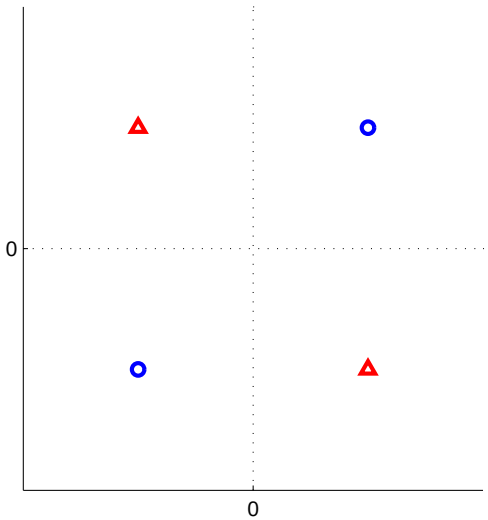
Non-separable Data



Solutions
Non-linear features, multiple perceptrons

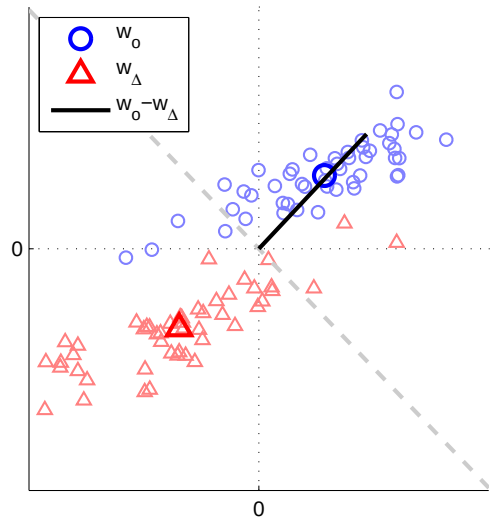
Problems with Linear Classifiers

Non-separable Data



Solutions
Non-linear features, multiple perceptrons

Correlated Data



Solution
Fisher's Linear Discriminant Analysis

Correlated Data?

In order to understand correlated data, it is helpful to

... know some real world examples:

1. The Iris data set
2. EEG data – Brain-Computer Interfaces (BCIs)

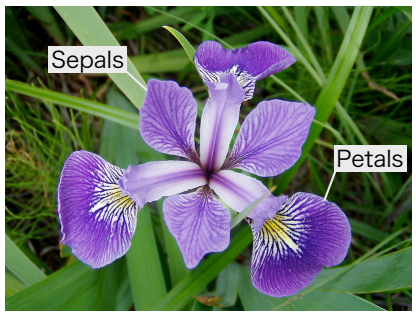
... know how it is generated

1. Univariate correlated data
2. Multivariate correlated data

The *Iris* Flower Dataset

Iris Setosa

Iris Versicolor

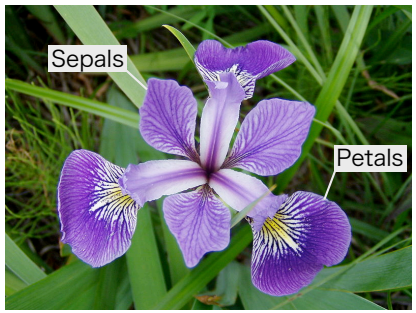


Iris Virginica

The *Iris* Flower Dataset

Iris Setosa

Iris Versicolor



Iris Virginica

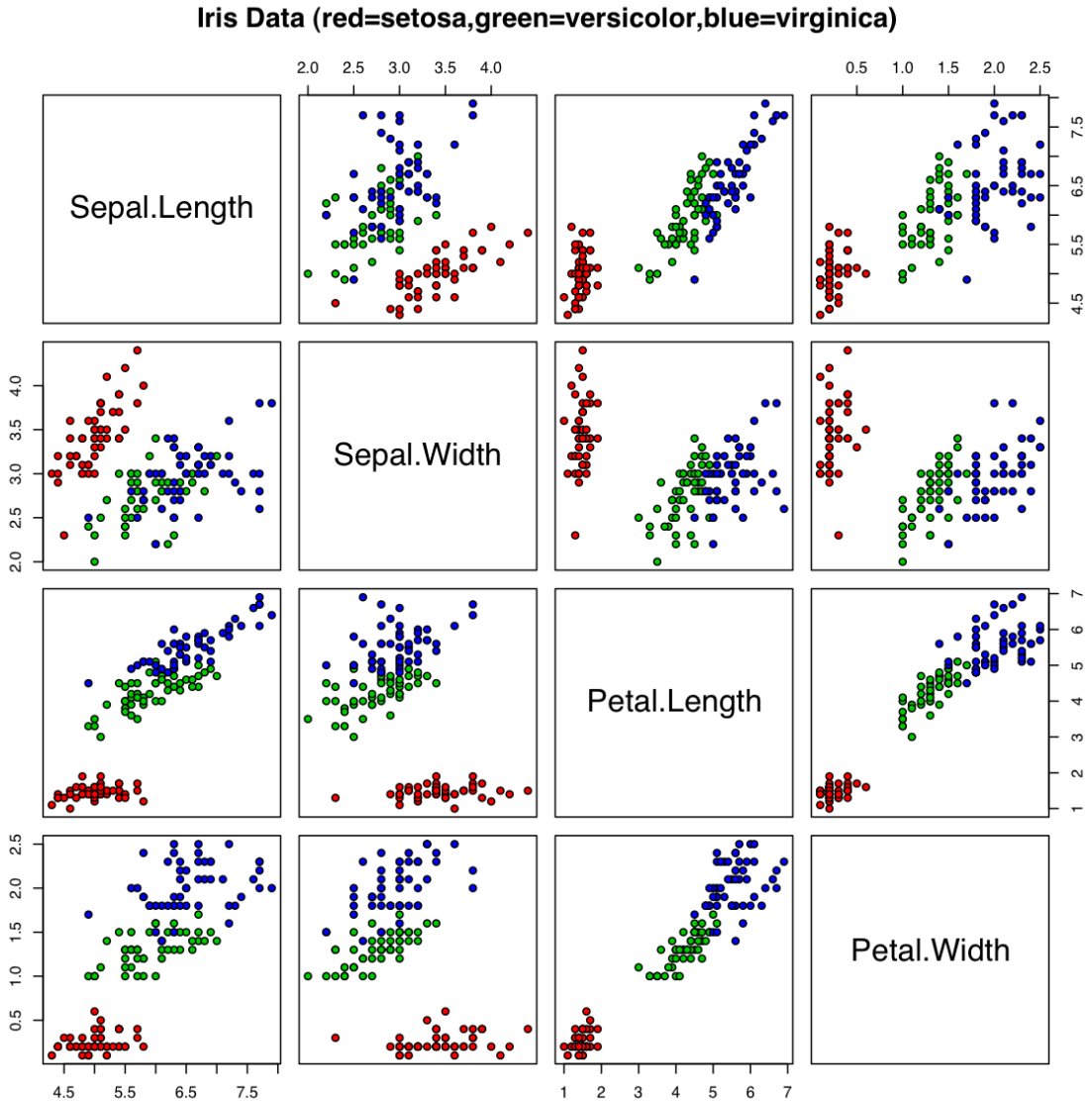
`http://en.wikipedia.org/wiki/Iris_flower_data_set`

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

Popular benchmark data set

The *Iris* Flower Dataset



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

Electroencephalogram (EEG) Data

A single neural current source $s(t)$ contributes linearly to the scalp potential $\mathbf{x}(t)$, i.e.,

$$\mathbf{x}(t) = \mathbf{a}s(t) \tag{1}$$

where $\mathbf{a} \in \mathbb{R}^D$ represents the coupling strengths of the source $\mathbf{s}$ to the D surface electrodes.

a depends on

- conductivity of brain tissue, skull, skin
- spatial location / orientation of current source in the brain
- impedances and locations of the scalp electrodes

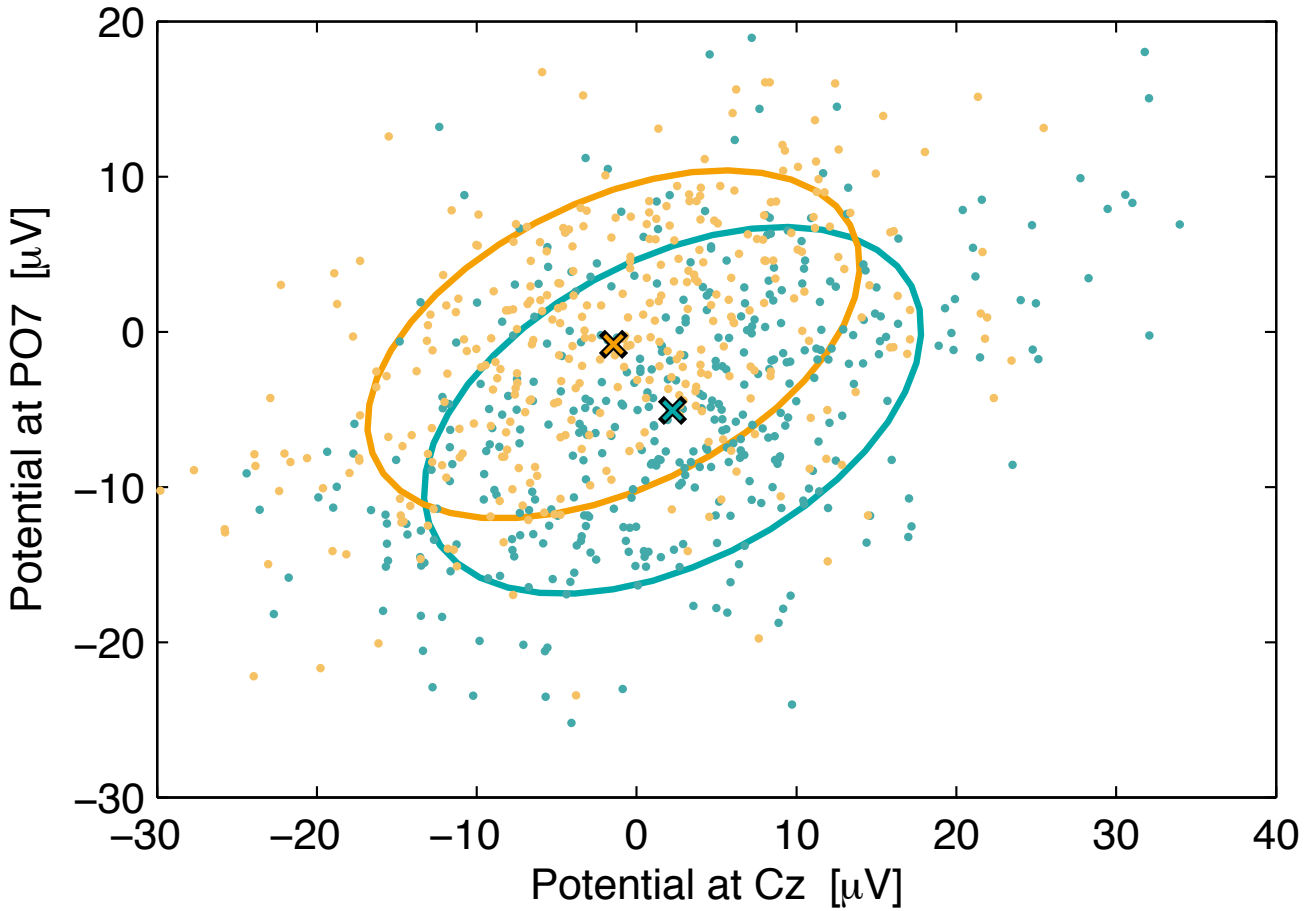
Electroencephalogram (EEG) Data

For more than one neural source the matrix form of eq. 1 is

$$X = AS \tag{2}$$

where X is a $D \times N$ matrix of EEG recordings and $S \in \mathbb{R}^{D \times N}$ is the matrix containing the D source time series.

Electroencephalogram (EEG) Data



Correlation Coefficients

Given $\mathbf{x} \in \mathbb{R}^{T \times 1}$ and $\mathbf{y} \in \mathbb{R}^{T \times 1}$ the empirical estimate of the **correlation coefficient** between $\mathbf{x}$ and $\mathbf{y}$ is

$$\text{Corr}(x, y) = \frac{\mathbf{x}^\top \mathbf{y}}{\sqrt{\mathbf{x}^\top \mathbf{x} \mathbf{y}^\top \mathbf{y}}} \tag{3}$$

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

Generating Univariate Correlated Data

A toy data experiment for univariate variables

$$\text{True Signal } x \sim \mathcal{N}(0,1), \qquad \text{Noise } \epsilon \sim \mathcal{N}(0,1)$$

$$\text{Measured Signal } y = \gamma\, x + \sqrt{1-\gamma^2}\, \epsilon$$

$$\rightarrow \text{Corr}(x,y) = \gamma$$

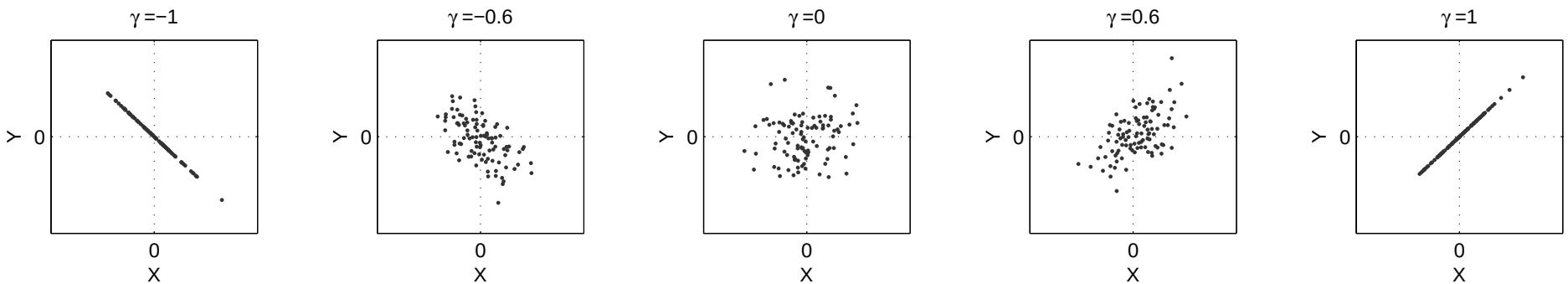
Generating Univariate Correlated Data

A toy data experiment for univariate variables

$$\text{True Signal } x \sim \mathcal{N}(0,1), \qquad \text{Noise } \epsilon \sim \mathcal{N}(0,1)$$

$$\text{Measured Signal } y = \gamma\, x + \sqrt{1-\gamma^2}\, \epsilon$$

$$\rightarrow \text{Corr}(x,y) = \gamma$$



Correlation Coefficients and Signal-to-Noise Ratio

A toy data experiment for univariate variables

$$\text{True Signal } x \sim \mathcal{N}(0,1),$$
$$\text{Noise } \epsilon \sim \mathcal{N}(0,1),$$
$$\text{where } \mathcal{N}(\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi)}} e^{-\sqrt{(\frac{x-\mu}{\sigma})^2}}$$
$$\text{Measured Signal } y = \gamma\ x + \sqrt{1-\gamma^2}\ \epsilon$$
$$\rightarrow \text{Signal-to-Noise Ratio} = \frac{\gamma^2}{1-\gamma^2}$$

Signal-to-Noise Ratio in Classification Settings

Consider one dimensional data $x \in \mathbb{R}^1$

N_+ data points from class $+$ and N_- data points from class $-$.

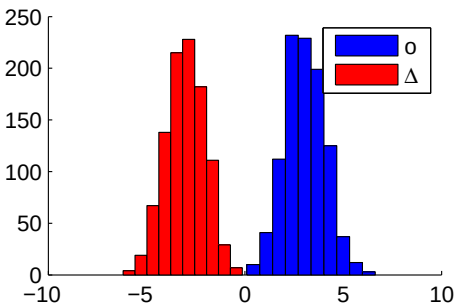
A useful definition of SNR for classification between $\mathbf{x}_+$ and $\mathbf{x}_-$ is

$$\frac{\mu_+ - \mu_-}{\sqrt{\sigma_+ + \sigma_-}} \tag{4}$$

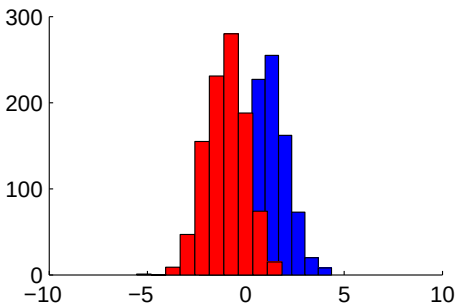
where $\mu_+ = 1/N_+ \sum_i^{N_+} \mathbf{x}_{i+}$ and $\sigma_+ = 1/N_+ \sum_i^{N_+} (\mathbf{x}_{i+} - \mu_+)^2$

The expression in eq. 4 is is sometimes called **t-value**

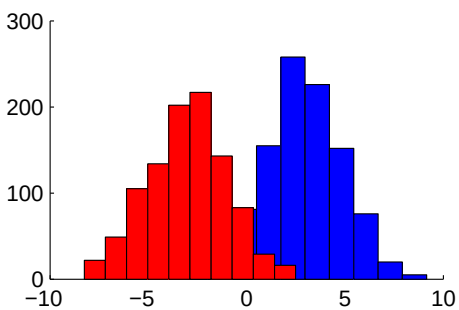
Good SNR



Bad SNR: Close means



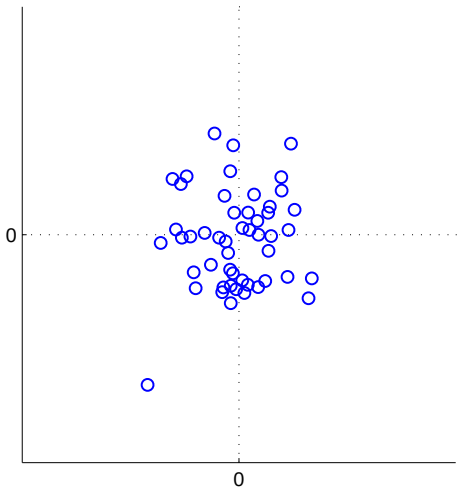
Bad SNR: Large Variance



Correlated Data and Linear Mappings

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

Uncorrelated



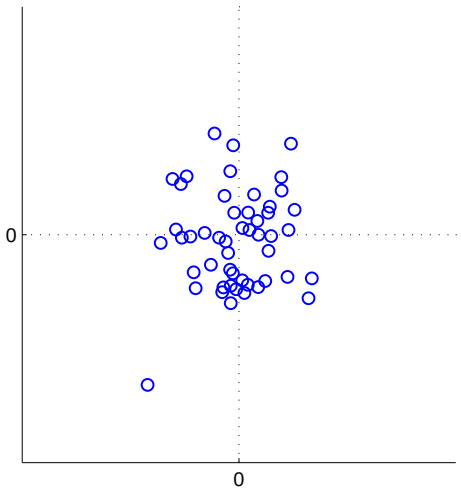
$$x \sim \mathcal{N}(0, 1)$$

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

Correlated Data and Linear Mappings

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

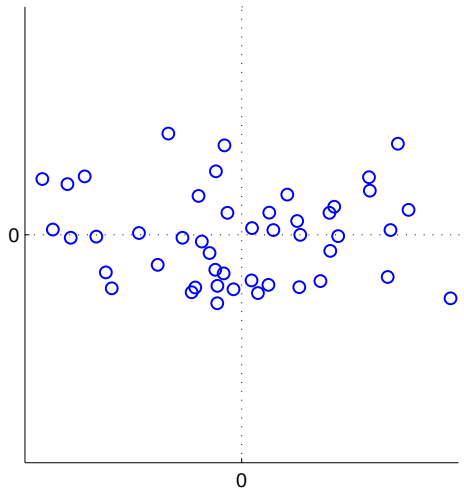
Uncorrelated



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

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

Uncorrelated, scaled

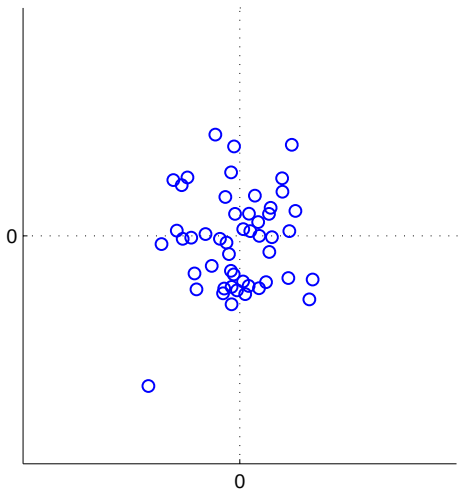


$$\begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix} X$$

Correlated Data and Linear Mappings

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

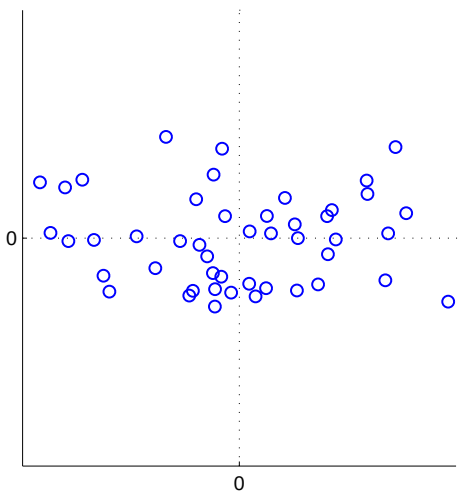
Uncorrelated



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

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

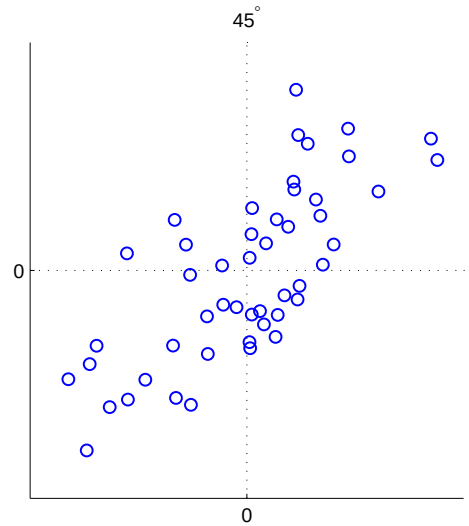
Uncorrelated, scaled



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

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

Scaled, rotated by 45°



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

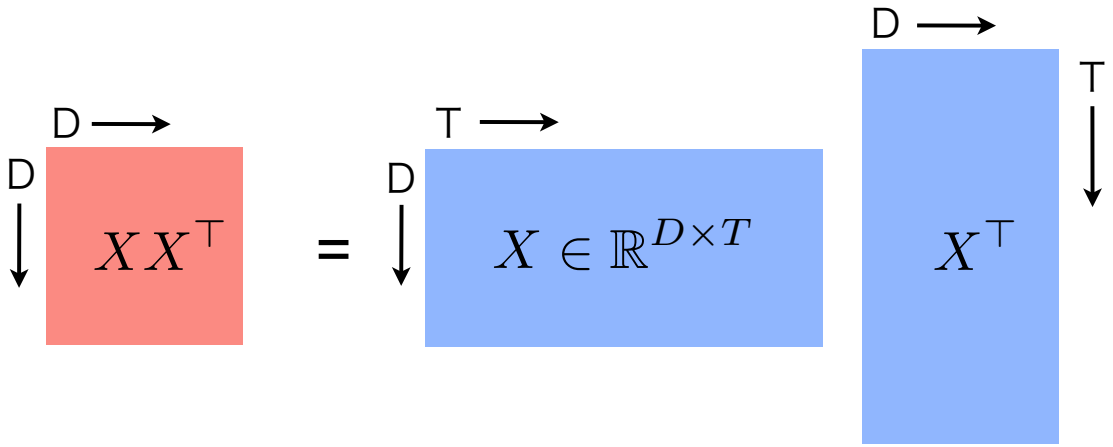
$$XX^\top = \begin{bmatrix} 5 & 4 \\ 4 & 5 \end{bmatrix}$$

Covariance Matrices

Given T data points $\mathbf{x} \in \mathbb{R}^D$ in a data matrix
$X = [\mathbf{x}_1, \ldots, \mathbf{x}_T] \in \mathbb{R}^{D \times T}$
the empirical estimate of the **covariance matrix** is defined as

$$1/T \;\; XX^\top \tag{5}$$

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



Generating Multivariate Correlated Data

How to generate Gaussian data for covariance matrix Σ?

Generating Multivariate Correlated Data

How to generate Gaussian data for covariance matrix Σ?

Let us assume data $X \in \mathbb{R}^{D \times N}$ was generated from uncorrelated Gaussian variables $S \sim \mathcal{N}(0, \mathbf{I})$
scaled by a diagonal matrix $\Lambda \in \mathbb{R}^{D \times D}$
rotated by a matrix $U \in \mathbb{R}^{D \times D}$

Generating Multivariate Correlated Data

How to generate Gaussian data for covariance matrix Σ?

$$\Sigma = XX^\top$$

Generating Multivariate Correlated Data

How to generate Gaussian data for covariance matrix Σ?

$$\begin{aligned}\Sigma &= XX^\top \\ &= U\Lambda \mathbf{I} \Lambda^\top U^\top\end{aligned}$$

Generating Multivariate Correlated Data

How to generate Gaussian data for covariance matrix Σ?

$$\begin{aligned}\Sigma &= XX^\top \\ &= U\Lambda \mathbf{I} \Lambda^\top U^\top \\ \Sigma &= U\Lambda^2 U^\top\end{aligned}$$

Generating Multivariate Correlated Data

How to generate Gaussian data for covariance matrix Σ?

$$
\begin{aligned}
\Sigma &= XX^\top \\
&= U\Lambda \mathbf{I} \Lambda^\top U^\top \\
\Sigma &= U\Lambda^2 U^\top \\
\Sigma U &= U\Lambda^2 \quad \rightarrow \text{ Eigen-Decomposition of } \Sigma
\end{aligned}
$$

Ronald A. Fisher

R.A. Fisher (1890 - 1962)

Founder of modern statistics

Interested in Biology

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

Fisher's Linear Discriminant Analysis

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

Maximizes mean class difference

$$(\mathbf{w}^\top \boldsymbol{\mu}_o - \mathbf{w}^\top \boldsymbol{\mu}_\Delta)^2 = \mathbf{w}^\top \ (\boldsymbol{\mu}_o - \boldsymbol{\mu}_\Delta)(\boldsymbol{\mu}_o - \boldsymbol{\mu}_\Delta)^\top \ \mathbf{w}$$

Minimizes variance in each class

$$\sum_i \left(\mathbf{w}^\top(\mathbf{x}_{oi} - \boldsymbol{\mu}_o)\right)^2 / N_o + \sum_j \left(\mathbf{w}^\top(\mathbf{x}_{\Delta j} - \boldsymbol{\mu}_\Delta)\right)^2 / N_\Delta$$

$$= \mathbf{w}^\top \left(1/N_o \sum_i (\mathbf{x}_{oi} - \boldsymbol{\mu}_o)(\mathbf{x}_{oi} - \boldsymbol{\mu}_o)^\top + 1/N_\Delta \sum_j (\mathbf{x}_{\Delta j} - \boldsymbol{\mu}_{\Delta j})(\mathbf{x}_{\Delta j} - \boldsymbol{\mu}_{\Delta j})^\top \right) \mathbf{w}$$

Fisher's Linear Discriminant Analysis

Maximization of $\mathbf{w}^\top S_B \mathbf{w}$ and simultaneous minimization of $\mathbf{w}^\top S_W \mathbf{w}$ is equivalent to maximizing the *Rayleigh quotient*

$$\underset{\mathbf{w}}{\operatorname{argmax}} \frac{\mathbf{w}^\top S_B \mathbf{w}}{\mathbf{w}^\top S_W \mathbf{w}} \tag{6}$$

where

$$S_B = (\boldsymbol{\mu}_o - \boldsymbol{\mu}_\Delta)(\boldsymbol{\mu}_o - \boldsymbol{\mu}_\Delta)^\top \tag{7}$$

$$S_W = 1/N_o \sum_i (\mathbf{x}_{oi} - \boldsymbol{\mu}_o)(\mathbf{x}_{oi} - \boldsymbol{\mu}_o)^\top + 1/N_\Delta \sum_j (\mathbf{x}_{\Delta j} - \boldsymbol{\mu}_{\Delta j})(\mathbf{x}_{\Delta j} - \boldsymbol{\mu}_{\Delta j})^\top \tag{8}$$

Note the similarity with the **t-value** in eq. 4
One can think of eq. 6 as a *multivariate t-value*

Fisher's Linear Discriminant Analysis

$$\underset{\mathbf{w}}{\text{argmax}} \frac{\mathbf{w}^\top S_B \mathbf{w}}{\mathbf{w}^\top S_W \mathbf{w}} \tag{6}$$

To optimize eq. 6 we set its derivative w.r.t w to 0

$$\frac{(\mathbf{w}^\top S_W \mathbf{w}) S_B \mathbf{w} - (\mathbf{w}^\top S_B \mathbf{w}) S_W \mathbf{w}}{(\mathbf{w}^\top S_W \mathbf{w})^2} = 0 \tag{7}$$

$$(\mathbf{w}^\top S_B \mathbf{w}) S_W \mathbf{w} = (\mathbf{w}^\top S_W \mathbf{w}) S_B \mathbf{w}$$

$$S_B \mathbf{w} = S_W \mathbf{w} \frac{\mathbf{w}^\top S_B \mathbf{w}}{\mathbf{w}^\top S_W \mathbf{w}}$$

Fisher's Linear Discriminant Analysis

$$\underset{\mathbf{w}}{\operatorname{argmax}} \frac{\mathbf{w}^\top S_B \mathbf{w}}{\mathbf{w}^\top S_W \mathbf{w}} \tag{6}$$

$$S_B \mathbf{w} = S_W \mathbf{w} \lambda$$

Note that left multiplying with $S_{\mathbf{w}}^{-1}$ yields

$$\mathbf{w} \propto S_{\mathbf{w}}^{-1}(\boldsymbol{\mu}_o - \boldsymbol{\mu}_\Delta)$$

$\rightarrow$ Fisher's LDA first *decorrelates* the data
followed by nearest centroid classification

A distance in decorrelated space is called **Mahalanobis Distance** [Mahalanobis, 1936]

Fisher's Linear Discriminant Algorithm

Algorithm 1 Fisher LDA - Two Classes

Require: Data $X = [\mathbf{x}_1, \ldots, \mathbf{x}_N], \mathbf{x}_i \in \mathbb{R}^D$, Labels $y_1, \ldots, y_N \in \{-1, +1\}$

Compute class mean vectors

$\boldsymbol{\mu}_{-1} = 1/|\mathcal{Y}_{-1}| \sum_{i \in \mathcal{Y}_{-1}} \mathbf{x}_i$

$\boldsymbol{\mu}_{+1} = 1/|\mathcal{Y}_{+1}| \sum_{j \in \mathcal{Y}_{+1}} \mathbf{x}_j$

Compute *between-class* covariance matrix

$S_B = (\boldsymbol{\mu}_{-1} - \boldsymbol{\mu}_{+1})(\boldsymbol{\mu}_{-1} - \boldsymbol{\mu}_{+1})^\top$

Compute *within-class* covariance matrices

$S_W = \sum_{i \in \mathcal{Y}_{-1}} (\mathbf{x}_i - \boldsymbol{\mu}_{-1})(\mathbf{x}_i - \boldsymbol{\mu}_{-1})^\top + \sum_{j \in \mathcal{Y}_{+1}} (\mathbf{x}_j - \boldsymbol{\mu}_{+1})(\mathbf{x}_j - \boldsymbol{\mu}_{+1})^\top$

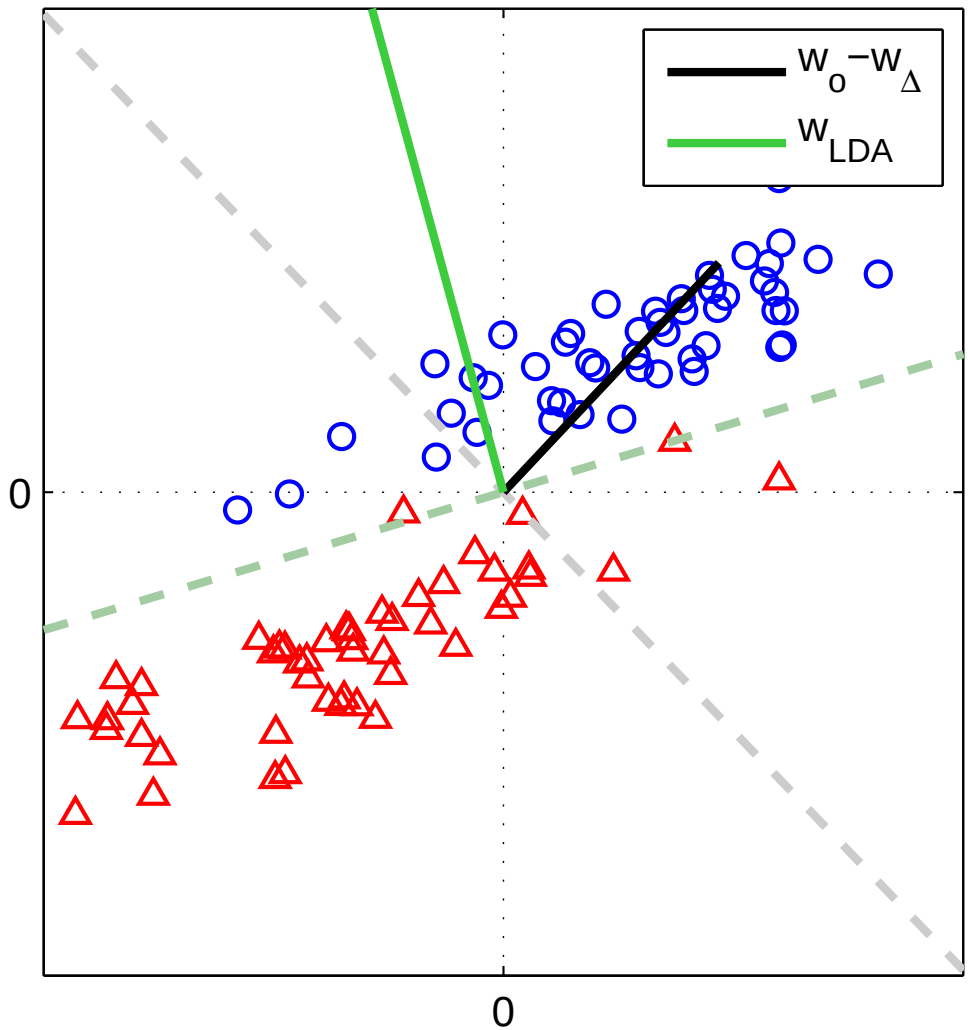
Compute eigenvalue decomposition

$S_B \mathbf{w} = S_W \mathbf{w} \lambda$

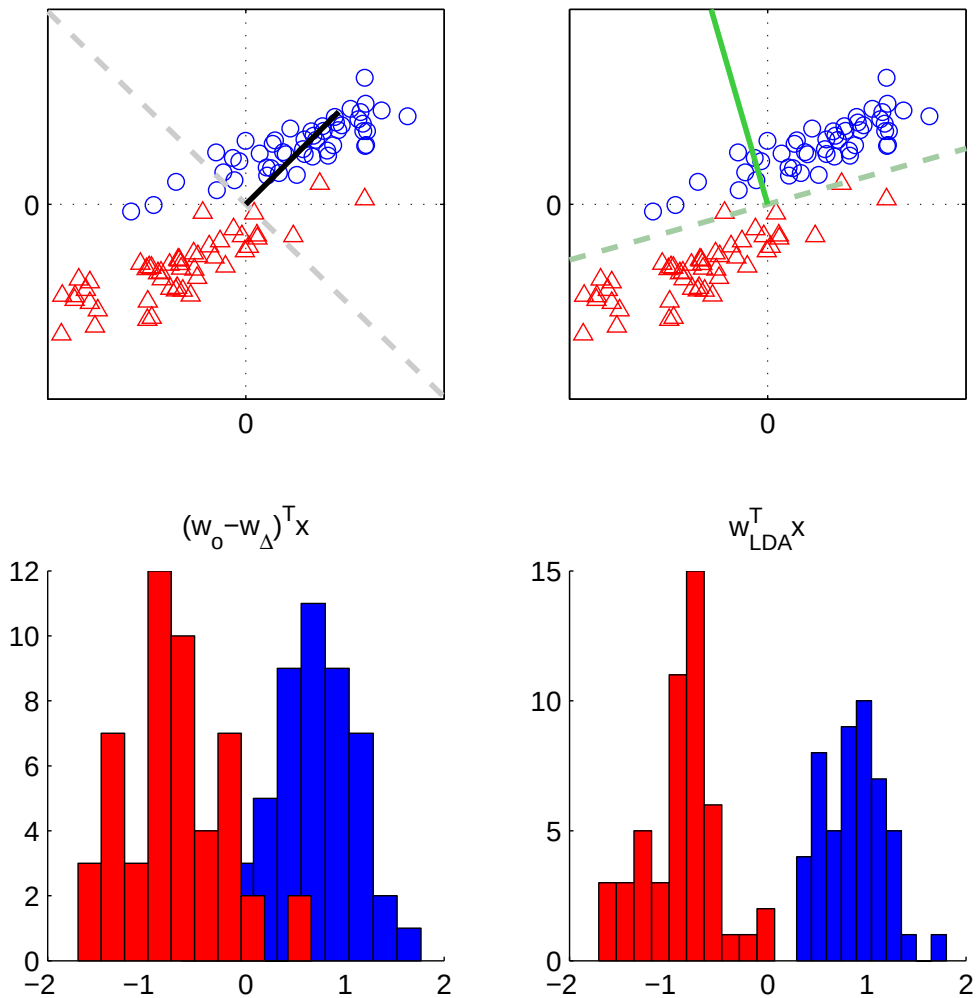
$\mathbf{b} = (\mathbf{w}^\top \boldsymbol{\mu}_{+1} + \mathbf{w}^\top \boldsymbol{\mu}_{-1})/2.$

return $\mathbf{w}, \mathbf{b}$

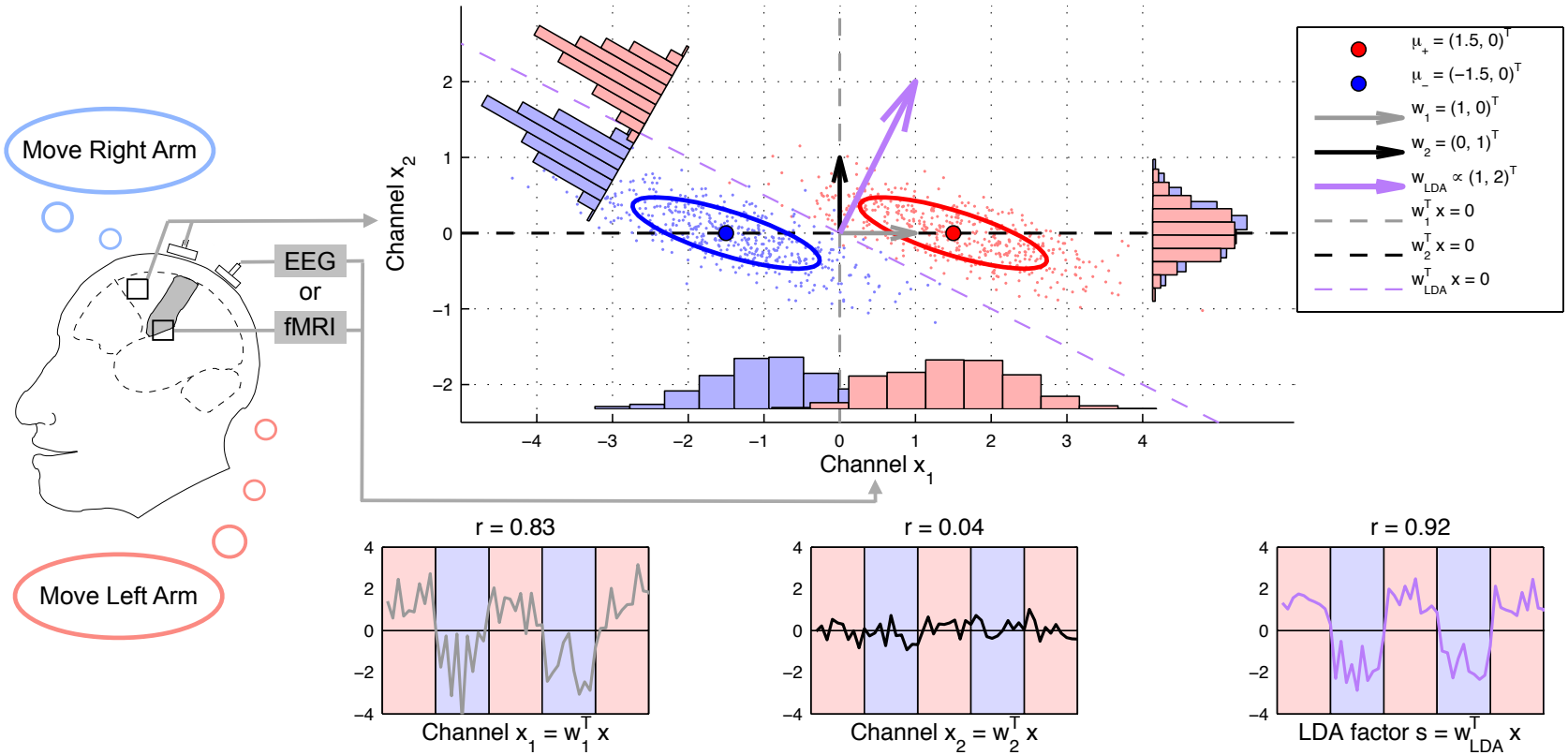
Fisher's LDA vs NCC



Fisher's Linear Discriminant Analysis



Fisher's Linear Discriminant Analysis



Problems with Fisher's LDA

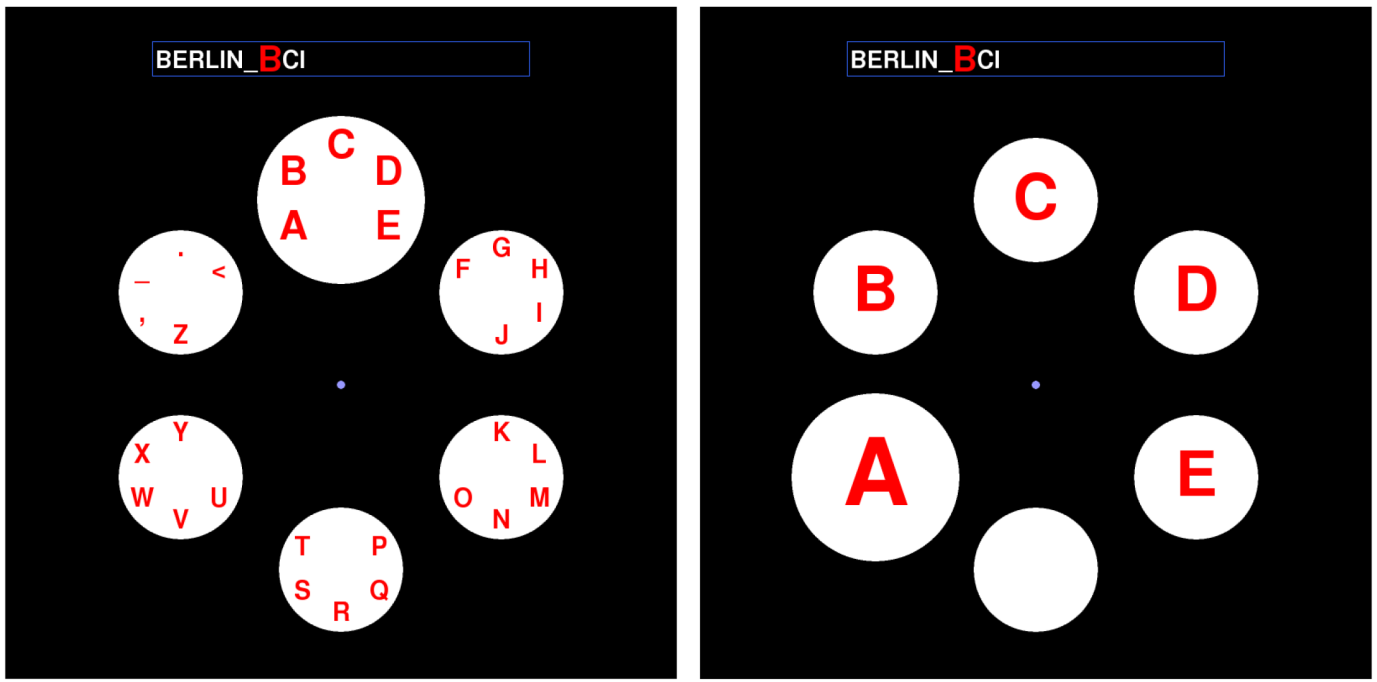
Objective Function in eq. 6 is a quadratic form
$\rightarrow$ LDA is not robust to outliers

Classifier outputs difficult to interpret
$\rightarrow$ Probabilistic outputs often desirable

Berlin Brain-Computer-Interface

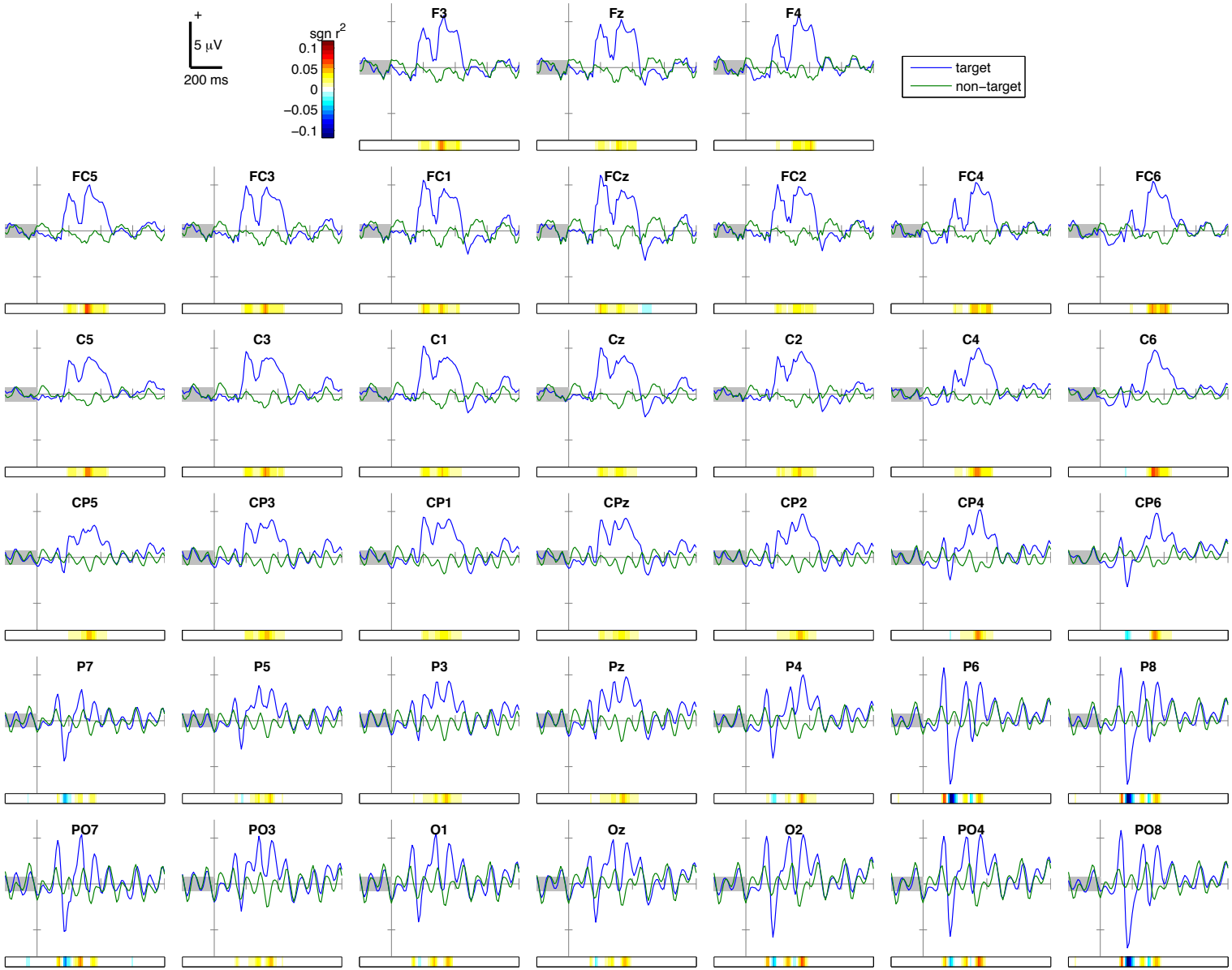
Hex-o-spell: Writing with thoughts

http://www.bbci.de/

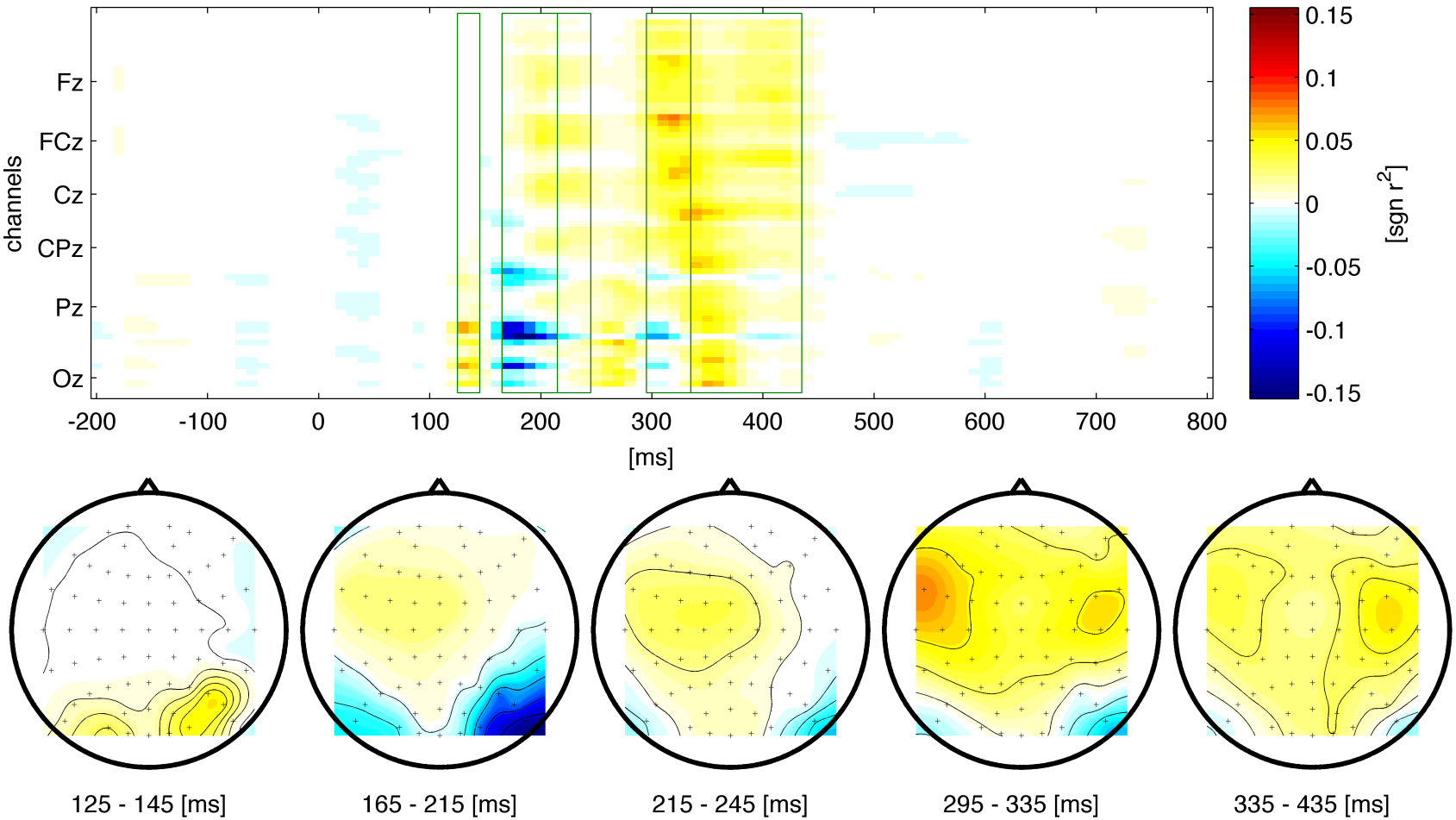


Demo: http://iopscience.iop.org/1741-2552/8/6/066003/media

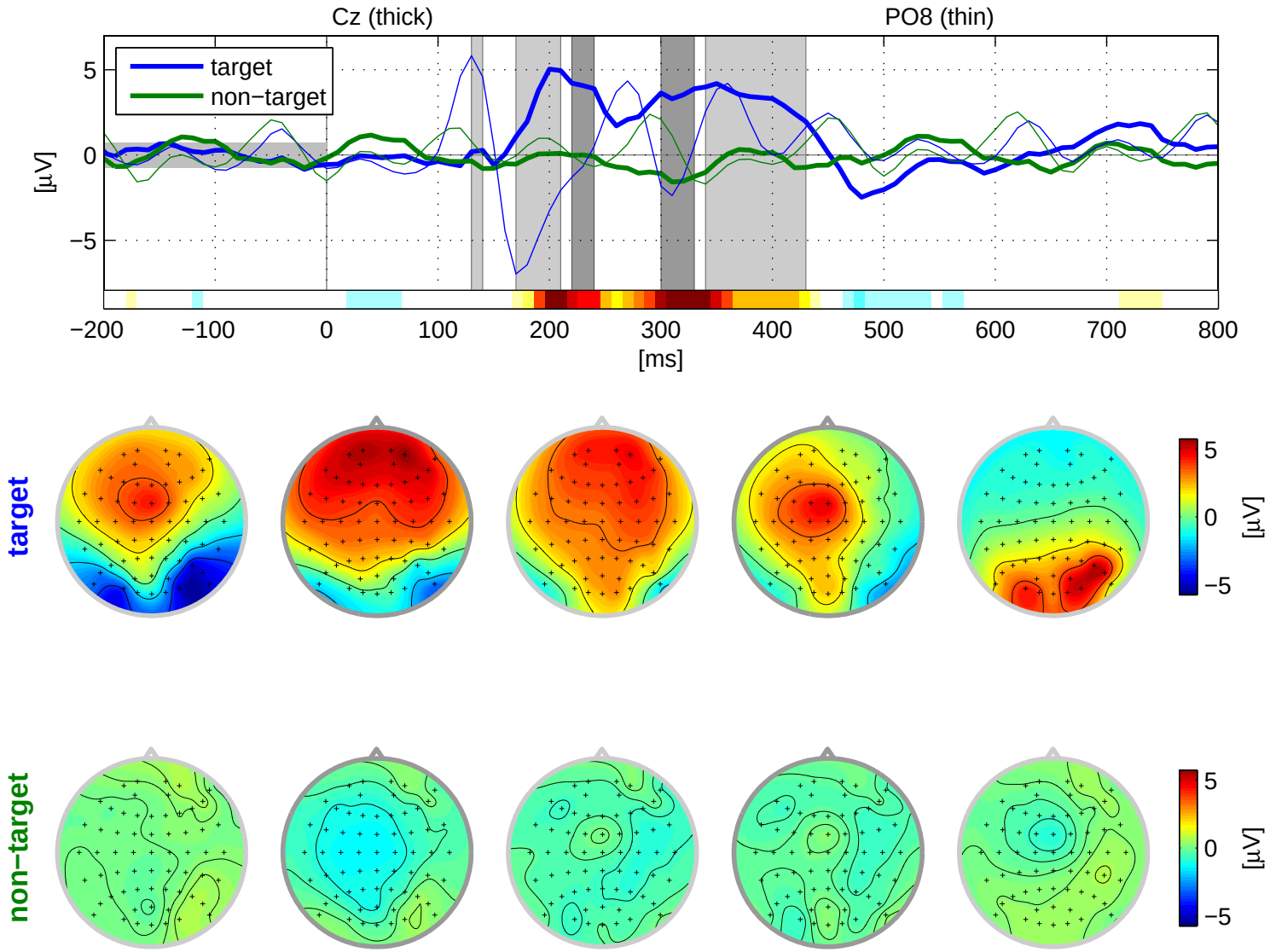
Scalp Potentials In Response to Targets/Non-Targets



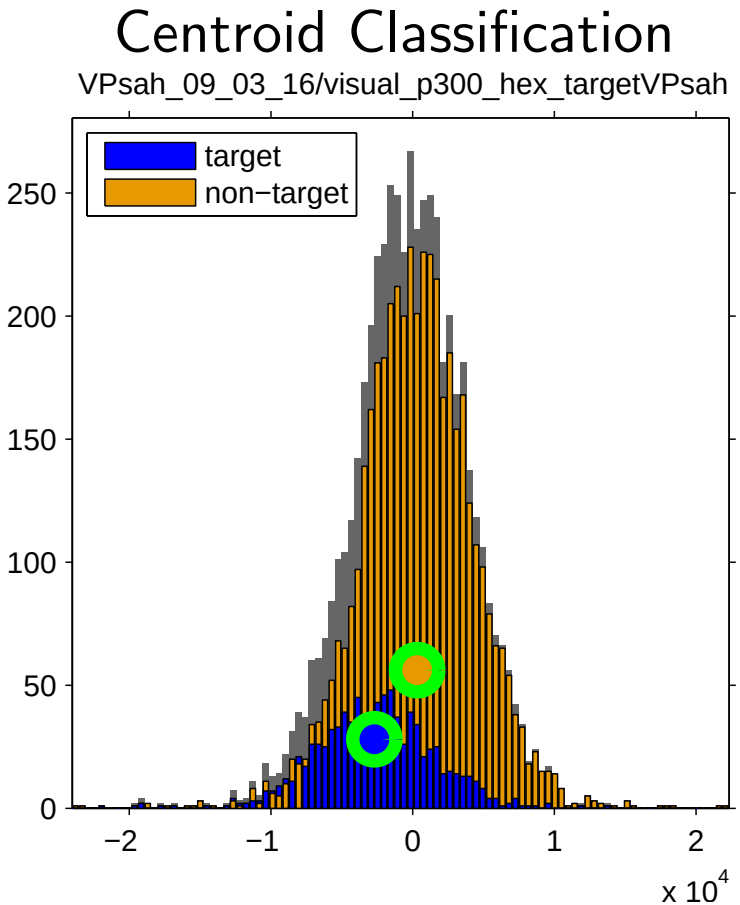
Berlin Brain-Computer-Interface



Berlin Brain-Computer-Interface

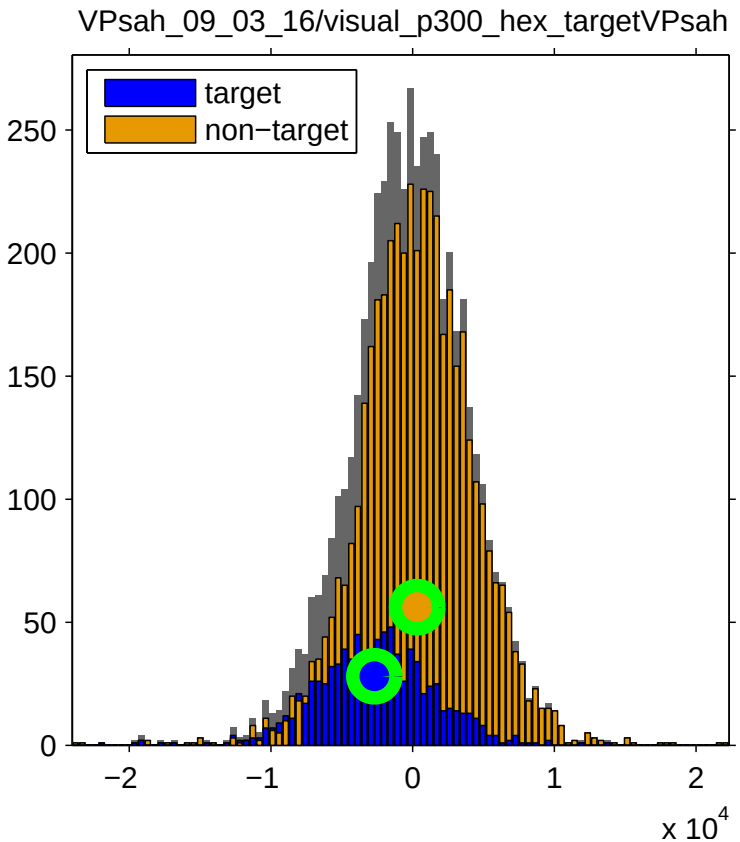


Berlin Brain-Computer-Interface

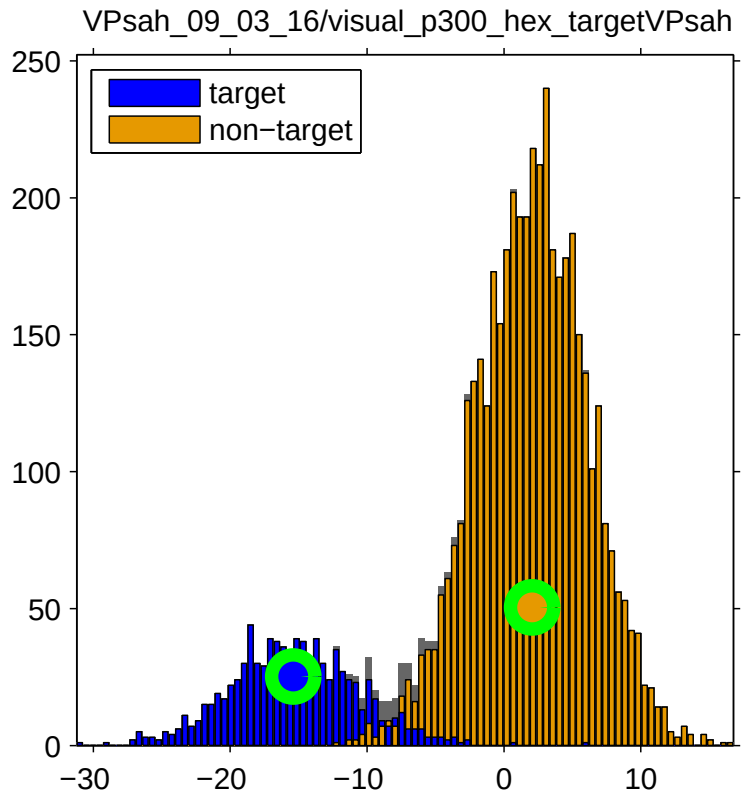


Berlin Brain-Computer-Interface

Centroid Classification



Fisher's LDA



Multiclass LDA

What if we have more than two classes?

LDA is nearest-centroid classification in a decorrelated space

$\mathbf{w} \propto S_{\mathbf{w}}^{-1}(\boldsymbol{\mu}_o - \boldsymbol{\mu}_\Delta)$

$\rightarrow$ Center and Decorrelate data in each class

$\rightarrow$ Project new data into decorrelated space

$\rightarrow$ Classify as class with closest centroid

Multiclass LDA

The centroids for each class k are

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{i=k} \mathbf{x}_i \tag{7}$$

where N_k is the number of data points in class k.

Multiclass LDA

The centroids for each class k are

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{i=k} \mathbf{x}_i \tag{7}$$

where N_k is the number of data points in class k.

The *between-class covariance matrix* $\mathbf{B} \in \mathbb{R}^{D \times D}$ as

$$\mathbf{B} = \sum_{k=1}^{K} (\boldsymbol{\mu}_k - \boldsymbol{\mu})(\boldsymbol{\mu}_k - \boldsymbol{\mu})^\top, \qquad \boldsymbol{\mu} = \sum_{i}^{N} \mathbf{x}_i \tag{8}$$

where $\boldsymbol{\mu}$ is the overall class mean.

Multiclass LDA

The centroids for each class k are

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{i=k} \mathbf{x}_i \tag{7}$$

where N_k is the number of data points in class k.

The *between-class covariance matrix* $\mathbf{B} \in \mathbb{R}^{D \times D}$ as

$$\mathbf{B} = \sum_{k=1}^{K} (\boldsymbol{\mu}_k - \boldsymbol{\mu})(\boldsymbol{\mu}_k - \boldsymbol{\mu})^\top, \qquad \boldsymbol{\mu} = \sum_{i}^{N} \mathbf{x}_i \tag{8}$$

where $\boldsymbol{\mu}$ is the overall class mean.

The *within-class covariance matrix* $\mathbf{S} \in \mathbb{R}^{D \times D}$ is

$$\mathbf{S} = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{N_k} \sum_{i}^{N_k} (\mathbf{x}_i - \boldsymbol{\mu}_k)(\mathbf{x}_i - \boldsymbol{\mu}_k)^\top \tag{9}$$

Multiclass LDA

Now the subspace $\mathbf{W}$ in which the data is optimally separated is

$$\underset{\mathbf{W}}{\operatorname{argmax}} \frac{\mathbf{W}^\top \mathbf{B} \mathbf{W}}{\mathbf{W}^\top \mathbf{S} \mathbf{W}}. \tag{10}$$

Multiclass LDA

Now the subspace $\mathbf{W}$ in which the data is optimally separated is

$$\underset{\mathbf{W}}{\operatorname{argmax}} \frac{\mathbf{W}^\top \mathbf{B} \mathbf{W}}{\mathbf{W}^\top \mathbf{S} \mathbf{W}}. \tag{10}$$

New data $\mathbf{x}$ belong to class with closest centroid $\mathbf{W}^\top \boldsymbol{\mu}$

$$k^* = \underset{k}{\operatorname{argmin}} \|\mathbf{W}^\top (\boldsymbol{\mu}_k - \mathbf{x})\|_2 \tag{11}$$

Multiclass LDA

Algorithm 2 Multiclass LDA - Finding Discriminative Subspace

Require: Data $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N] \in \mathbb{R}^{D \times N}$, Labels $y_1, \ldots, y_N \in \{1, \ldots, K\}$
Ensure: Discriminative Subspace $\mathbf{W}$, class means $[\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k]$

```
# Compute overall mean
μ = 1/N ∑_i^N x_i
for Class k = 1, ..., K do
    # Compute class mean vectors
    μ_k = 1/N_k ∑_i^{N_k} x_i
    # Compute within-class covariance matrix
    S_k = 1/N_k ∑_i^{N_k} (x_i − μ_k)(x_i − μ_k)^⊤
end for
# Compute between-class covariance matrix B
B = ∑_{k=1}^K (μ_k − μ)(μ_k − μ)^⊤
# Compute within-class covariance matrix S
S = ∑_{k=1}^K S_k
# Compute the first k − 1 eigenvalues
BW = SWΛ
```

Multiclass LDA

Algorithm 3 Multiclass Fisher LDA - Predictions with new data

Require: Data point $\mathbf{x} \in \mathbb{R}^D$, $\mathbf{W} \in \mathbb{R}^{D \times K}$, $\boldsymbol{\mu}_k,\ k \in \{1, \ldots, K\}$

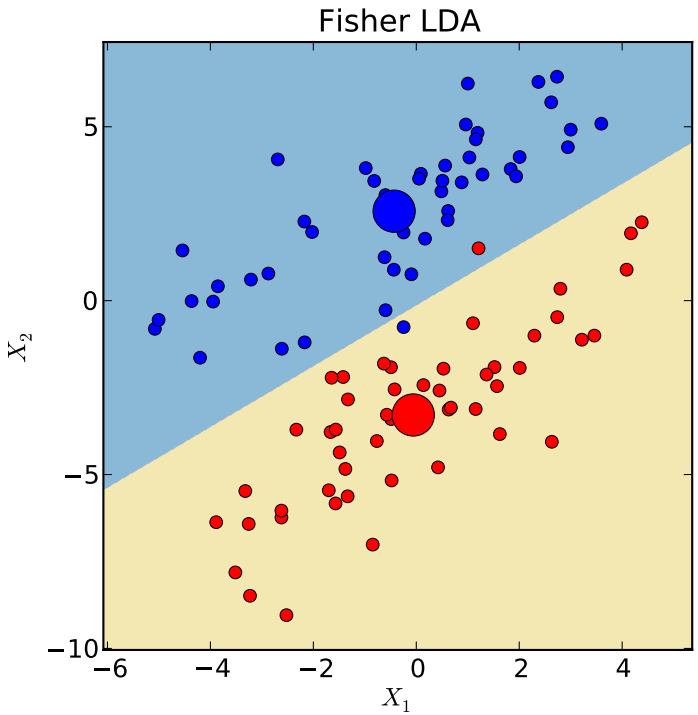
Ensure: Class membership k^*

1: # Compute nearest class centroid in discriminative subspace

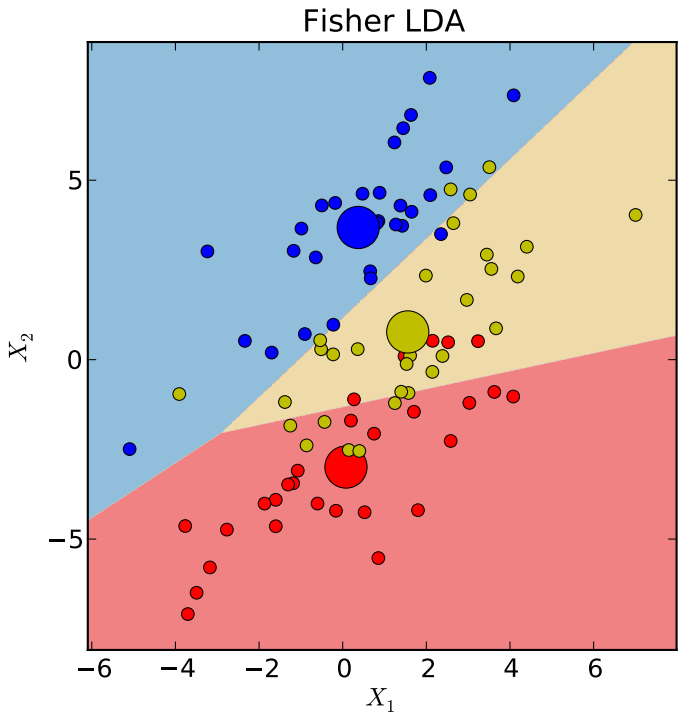
2: $k^* = \text{argmin}_k \|\mathbf{W}^\top(\boldsymbol{\mu}_k - \mathbf{x})\|_2$.

Multiclass LDA

Two-Class Problem



Three-Class Problem



Bias in Estimating Covariance Matrices

For LDA we need estimates for the distribution parameters:

- $\hat{\mu} = \frac{1}{n}\sum_{k=1}^{n} \mathbf{x}_k$ **empirical mean**
- $\hat{\Sigma} = \frac{1}{n-1}\sum_{k=1}^{n}(\mathbf{x}_k - \hat{\mu})(\mathbf{x}_k - \hat{\mu})^\top$ **emp. covariance matrix**

But, if the number of samples n is not large relative to the dimension d, the estimation, in particular $\hat{\Sigma}$, is error-prone.

This may affect classification with LDA badly.

There is a systematical bias in the empirical covariance matrix:

- Large Eigenvalues of $\hat{\Sigma}$ are too large
- Small Eigenvalues of $\hat{\Sigma}$ are too small

assuming $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$ are drawn from $\mathcal{N}(\mu, \Sigma)$.

Bias in Estimating Covariance Matrices

For LDA we need estimates for the distribution parameters:

- $\hat{\boldsymbol{\mu}} = \frac{1}{n}\sum_{k=1}^{n} \mathbf{x}_k$ **empirical mean**
- $\hat{\Sigma} = \frac{1}{n-1}\sum_{k=1}^{n}(\mathbf{x}_k - \hat{\boldsymbol{\mu}})(\mathbf{x}_k - \hat{\boldsymbol{\mu}})^\top$ **emp. covariance matrix**

But, if the number of samples n is not large relative to the dimension d, the estimation, in particular $\hat{\Sigma}$, is error-prone.

This may affect classification with LDA badly.

There is a systematical bias in the empirical covariance matrix:

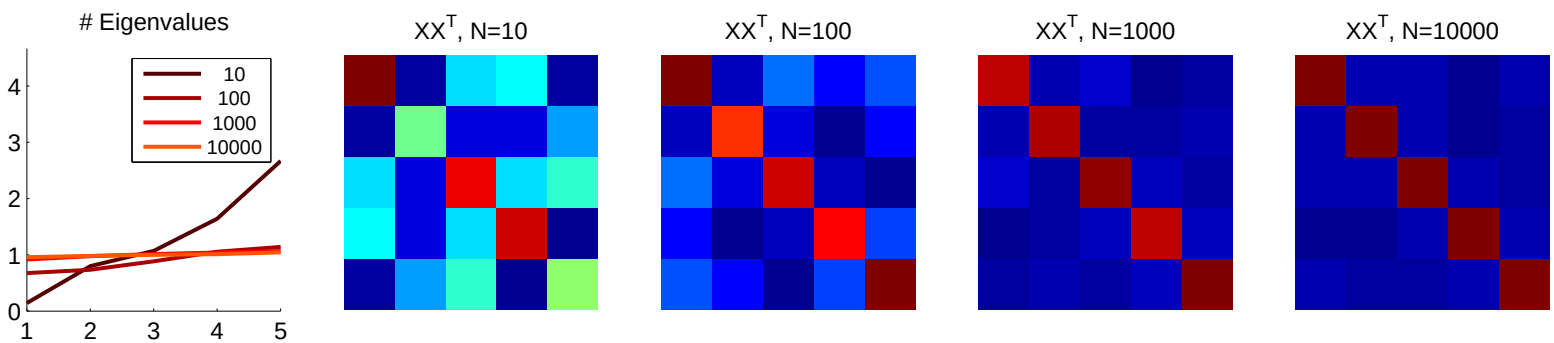
- Large Eigenvalues of $\hat{\Sigma}$ are too large
- Small Eigenvalues of $\hat{\Sigma}$ are too small

assuming $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$ are drawn from $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$.

Bias in Estimating Covariances (2)

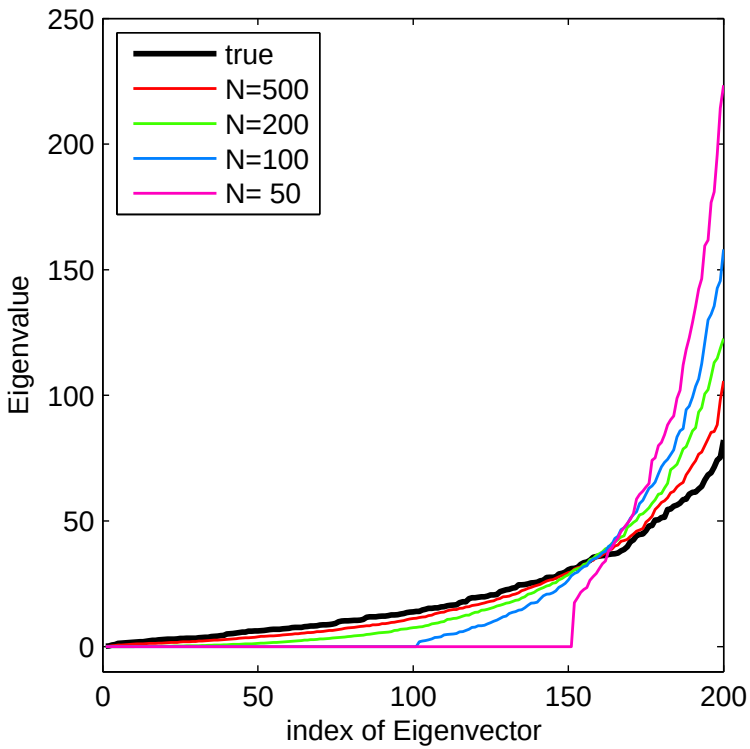
Example with uncorrelated gaussian data

$$\mathbf{x} \sim \mathcal{N}(0, \mathbf{I}),\ \mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N],\ \mathbb{E}[\mathbf{X}\mathbf{X}^\top] = \mathbf{I}$$

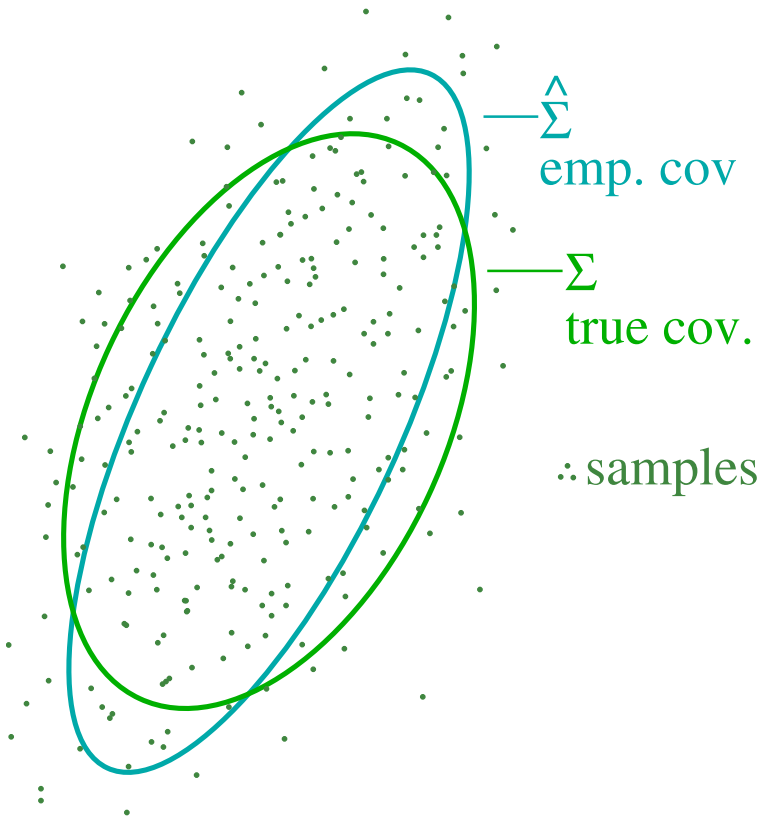


Bias in Estimating Covariances (3)

Simulation for $d = 200$:



Cartoon in 2D:

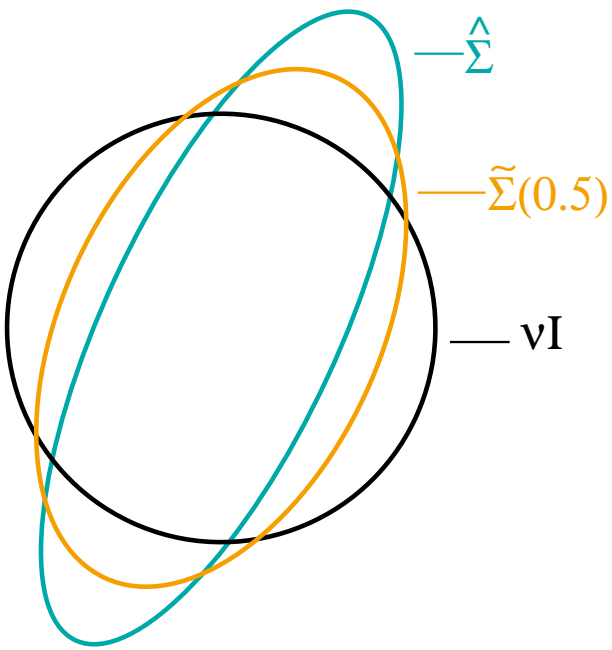


A Remedy for the Estimation Bias

A simple way that counteracts the bias is **shrinkage**:
The empirical covariance matrix $\hat{\Sigma}$ is modified to be more spherical:

$$\tilde{\Sigma}(\gamma) = (1-\gamma)\hat{\Sigma} + \gamma\nu\mathbf{I}$$

for a $\gamma \in [0, 1]$ and ν defined as average Eigenvalue trace$(\hat{\Sigma})/d$.



Next, we check that shrinkage serves the intended purpose. Covariance matrices are described by their Eigenvectors and Eigenvalues. So, we have to investigate, what happens to those, when we change over from the empirical covariance matrix $\hat{\Sigma}$.

Properties of the Shrunk Covariance Matrix

From the Eigenvalue decomposition of the empirical covariance matrix $\hat{\Sigma} = VDV^\top$ with orthonormal V and diagonal D, we get an Eigenvalue decomposition of $\tilde{\Sigma}(\gamma) = (1-\gamma)\hat{\Sigma} + \gamma\nu\mathbf{I}$ like this:

$$\begin{aligned}\tilde{\Sigma}(\gamma) &= (1-\gamma)VDV^\top + \gamma\nu\mathbf{I} \\ &= (1-\gamma)VDV^\top + \gamma\nu V\mathbf{I}V^\top \\ &= V\underbrace{\left((1-\gamma)D + \gamma\nu\mathbf{I}\right)}_{\text{diagonal matrix}}V^\top\end{aligned}$$

We see that

- $\hat{\Sigma}$ and $\tilde{\Sigma}(\gamma)$ have the same Eigenvectors (columns of V)
- Extreme Eigenvalues (large/small) are shrunk/extended towards the average Eigenvalue ν as $d_i \mapsto (1-\gamma)d_i + \gamma\nu$
- $\gamma = 0$ means no shrinkage: $\tilde{\Sigma}(0) = \hat{\Sigma}$
- $\gamma = 1$ corresponds to spherical covariances matrices: $\tilde{\Sigma}(1) = \nu\mathbf{I}$

Properties of the Shrunk Covariance Matrix

From the Eigenvalue decomposition of the empirical covariance matrix $\hat{\Sigma} = VDV^\top$ with orthonormal V and diagonal D, we get an Eigenvalue decomposition of $\tilde{\Sigma}(\gamma) = (1-\gamma)\hat{\Sigma} + \gamma\nu\mathbf{I}$ like this:

$$\begin{aligned}\tilde{\Sigma}(\gamma) &= (1-\gamma)VDV^\top + \gamma\nu\mathbf{I} \\ &= (1-\gamma)VDV^\top + \gamma\nu V\mathbf{I}V^\top \\ &= V\underbrace{\left((1-\gamma)D + \gamma\nu\mathbf{I}\right)}_{\text{diagonal matrix}}V^\top\end{aligned}$$

We see that

- $\hat{\Sigma}$ and $\tilde{\Sigma}(\gamma)$ have the same Eigenvectors (columns of V)
- Extreme Eigenvalues (large/small) are shrunk/extended towards the average Eigenvalue ν as $d_i \mapsto (1-\gamma)d_i + \gamma\nu$
- $\gamma = 0$ means no shrinkage: $\tilde{\Sigma}(0) = \hat{\Sigma}$
- $\gamma = 1$ corresponds to spherical covariances matrices: $\tilde{\Sigma}(1) = \nu\mathbf{I}$

Regularized Linear Discriminant Analysis

This technique can be used to enhance LDA to work better in the case of a low number-of-samples to dimensionality ratio. The empirical covariance matrix $\hat{\Sigma}$ is replaced by a shrunk covariance matrix $\tilde{\Sigma}(\gamma)$:

$$\mathbf{w}_\gamma := \tilde{\Sigma}(\gamma)^{-1}(\mu_2 - \mu_1)$$

Here, γ is a hyper parameter that has to be selected between 0 and 1.

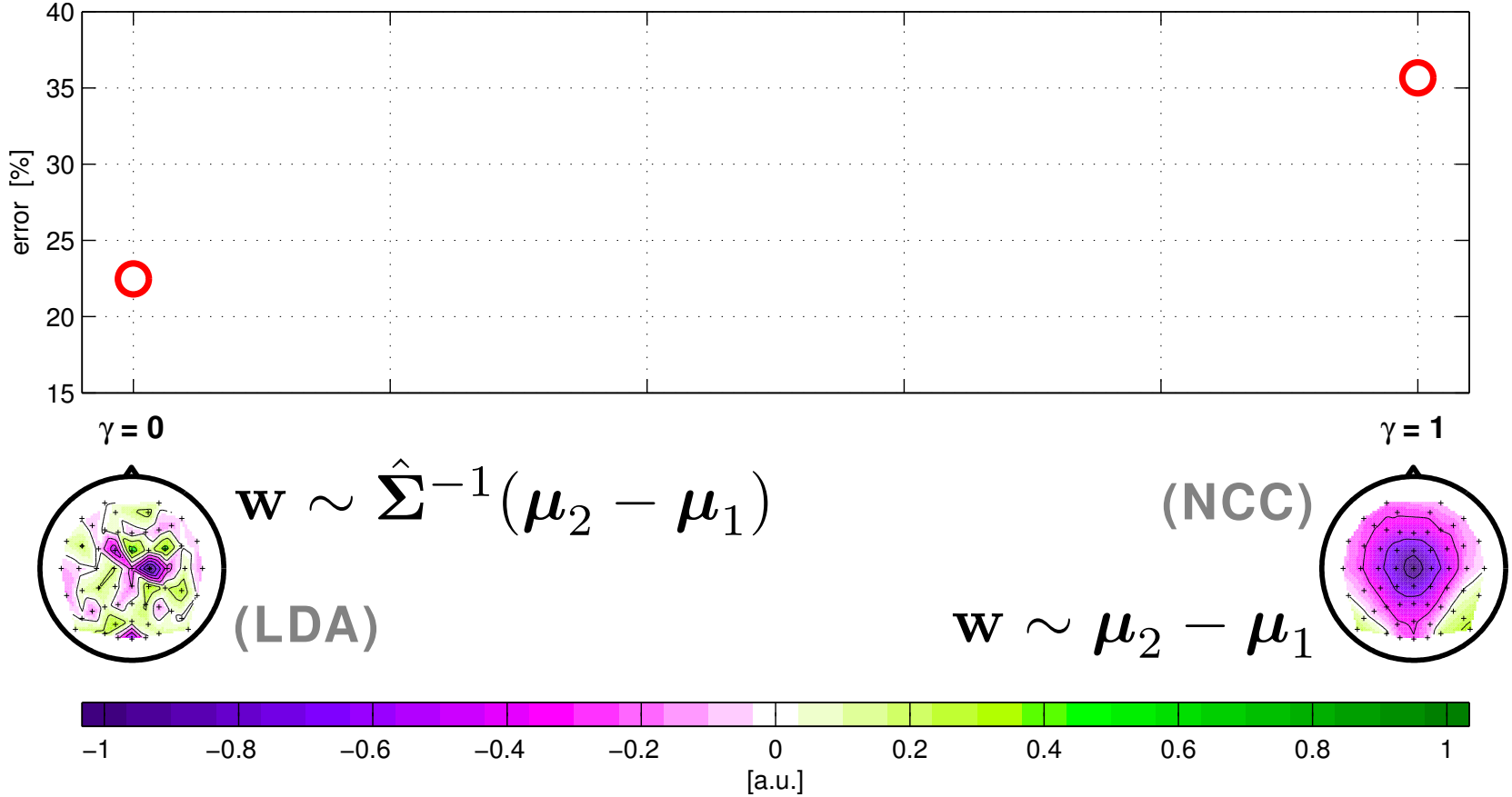
- $\gamma = 0$ yields $\mathbf{w}_0 = \hat{\Sigma}^{-1}(\mu_2 - \mu_1)$, i.e. unregularized LDA
- $\gamma = 1$ yields $\mathbf{w}_1 = \mu_2 - \mu_1$, i.e. NCC

Before addressing the choice of γ, let us look at the impact of the shrinkage parameter.

Impact of Shrinkage as Trade-off

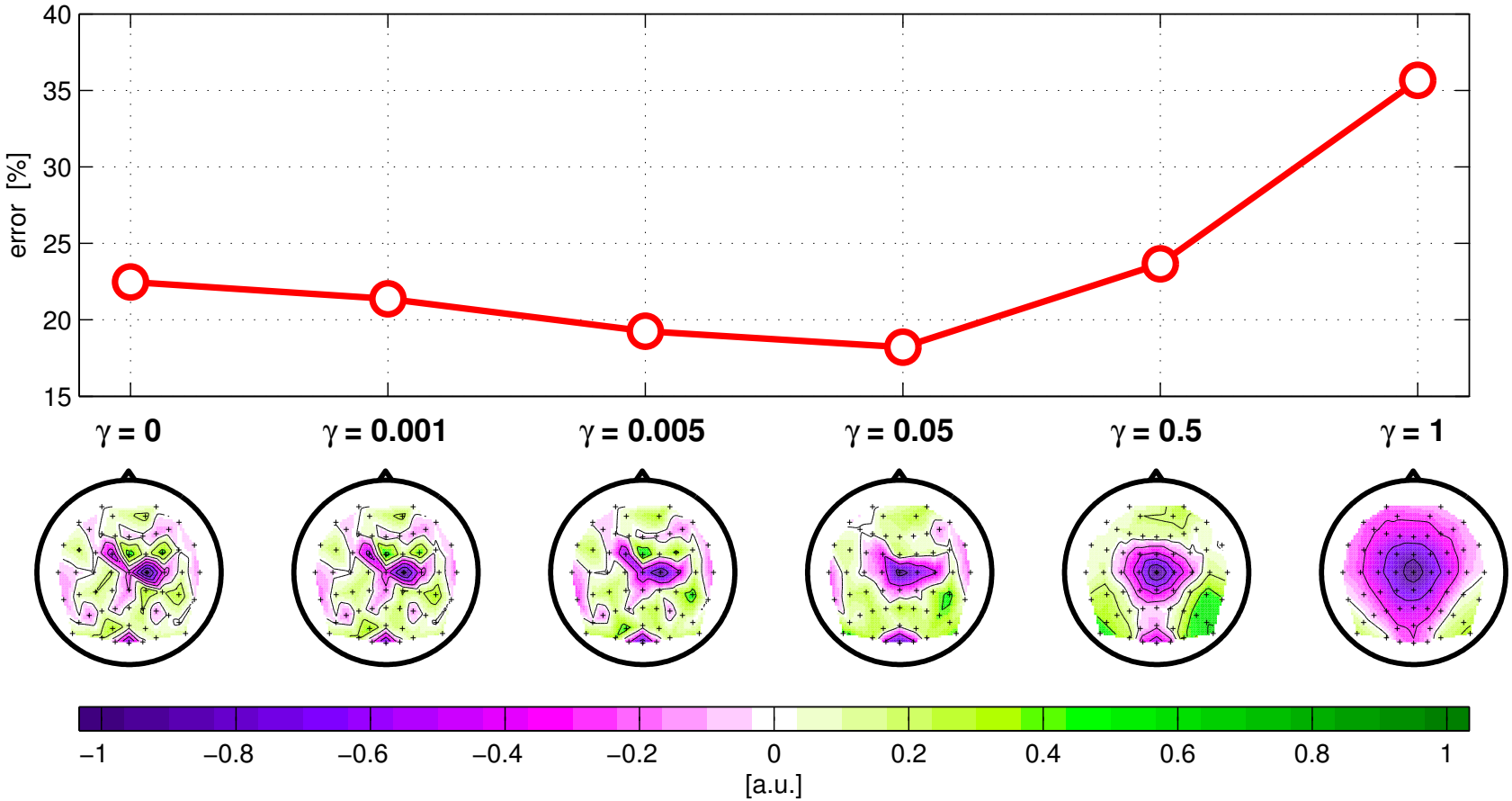
LDA with shrinkage: $\mathbf{w} = \tilde{\Sigma}(\gamma)^{-1}(\mu_2 - \mu_1)$;

$\tilde{\Sigma}(\gamma) = (1-\gamma)\hat{\Sigma} + \gamma\nu\mathbf{I}$



Impact of Shrinkage as Trade-off

With increasing shrinkage, the spatial filters (classifier) look smoother, but classification may degrade with too much shrinkage.



So How Can We Determine Shrinkage Parameters?

Two ways of determining the shrinkage parameter γ are popular

- Cross-Validation (Trial-and-Error)
  Try different γ's and test the error on a **hold-out data set**
- Analytical Solution
  Calculate the optimal γ [Ledoit and Wolf, 2004]

Cross-Validation

Split data set in F different **training** and **test** data folds

fold 1 [$\underbrace{x_1,\ x_2,\ x_3,\ x_4}_{\mathcal{F}_1^{\text{train}}}, \underbrace{x_5,\ x_6}_{\mathcal{F}_1^{\text{test}}}$]

fold 2 [$\underbrace{x_1,\ x_2,}_{\mathcal{F}_1^{\text{test}}}\ \underbrace{x_3,\ x_4,\ x_5,\ x_6}_{\mathcal{F}_1^{\text{train}}}$]

fold 3 ...

For each fold:

Train your model on the training data on $\mathcal{F}^{\text{train}}$

Test your model on the test data on $\mathcal{F}^{\text{test}}$

Cross-Validation Algorithm

Algorithm 4 Cross-Validation

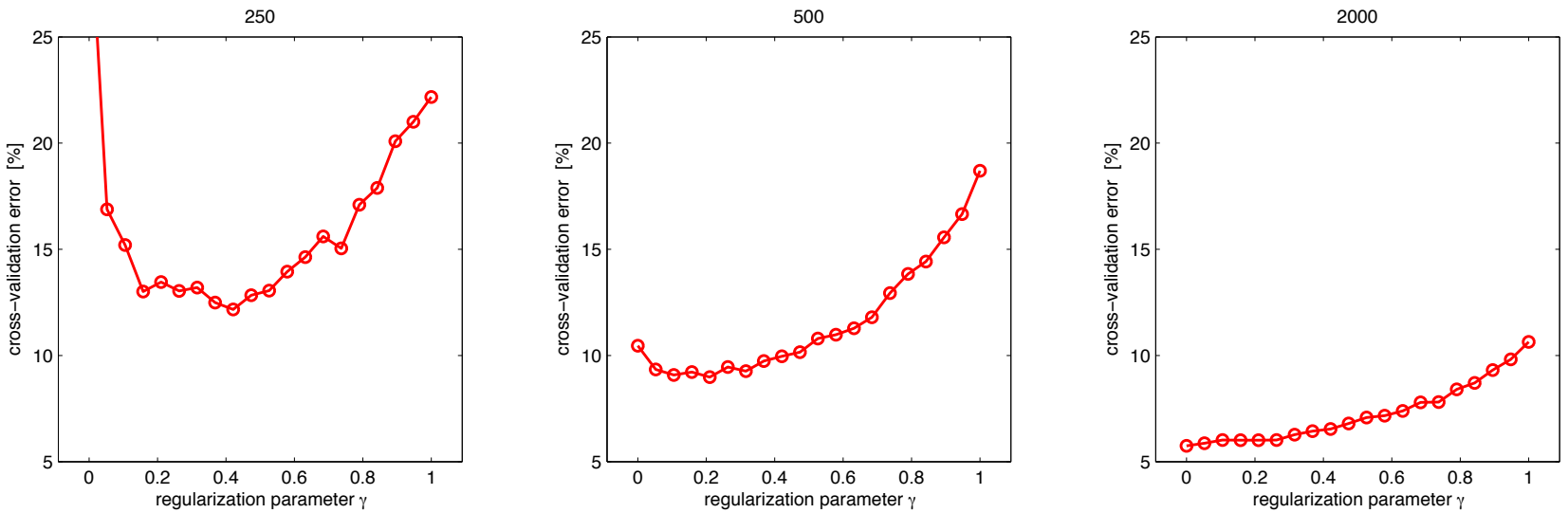
Require: Data $\{(\mathbf{x}_1, y_1) \ldots, (\mathbf{x}_N, y_N)\}$, Number of CV folds F

Ensure: Average Generalization Prediction Accuracy

1: # Split data in F **non-overlapping** folds
2: **for** Fold $f = 1, \ldots, F$ **do**
3: # '\' denotes Set Difference
4: # Train model on folds $\{1, \ldots, F\} \setminus f$
5: # Compute prediction on test data fold f
6: **end for**
7: # Compute Average Prediction Accuracy over all test data sets

LDA with Different Shrinkage Parameters

Cross-validation results for different sizes of training data (250, 500, 2000) for different values of the shrinkage parameter γ (x-axis). Features vectors have 250 dimensions.



Few data (relative to dimensions)

$\Rightarrow$ Bad Covariance Matrix estimates

$\Rightarrow$ More regularization (higher γ) needed!

Summary

- NCC ignores **covariance** structure of data
- This can lead to poor classification performance
- (Fisher) Linear Discriminant Analysis accounts for covariance
- Fisher LDA requires to estimate covariance matrices
- Few data points $\Rightarrow$ bad covariance estimates
- Regularization can improve covariance estimates
- Easiest regularization: **shrinkage** $\tilde{\Sigma} = (1-\gamma)\hat{\Sigma} + \gamma\nu\mathbf{I}$
- Optimal regularization parameters γ found using cross-validation

References

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

O. Ledoit and M. Wolf. A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis*, 88(2):365 – 411, 2004.

P. Mahalanobis. On the generalized distance in statistics. *Proc. Nat. Inst. Sci. India (Calcutta)*, 2:49–55, 1936.