

## Cognitive Algorithms Lecture 2

# Linear Classification

Klaus-Robert Müller, Johannes Niediek,  
Augustin Krause, Joanina Oltersdorff, Ken Schreiber

Technische Universität Berlin  
Machine Learning Group

## Cognitive Algorithms Lecture 2

# Linear Classification

Klaus-Robert Müller, Johannes Niediek,  
Augustin Krause, Joanina Oltersdorff, Ken Schreiber

Technische Universität Berlin  
Machine Learning Group

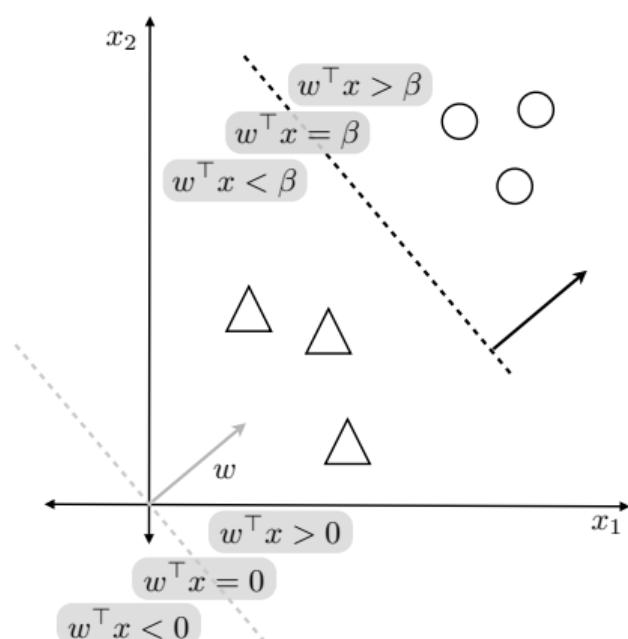
## Cognitive Algorithms Lecture 2

# Linear Classification

Klaus-Robert Müller, Johannes Niediek,  
Augustin Krause, Joanina Oltersdorff, Ken Schreiber

Technische Universität Berlin  
Machine Learning Group

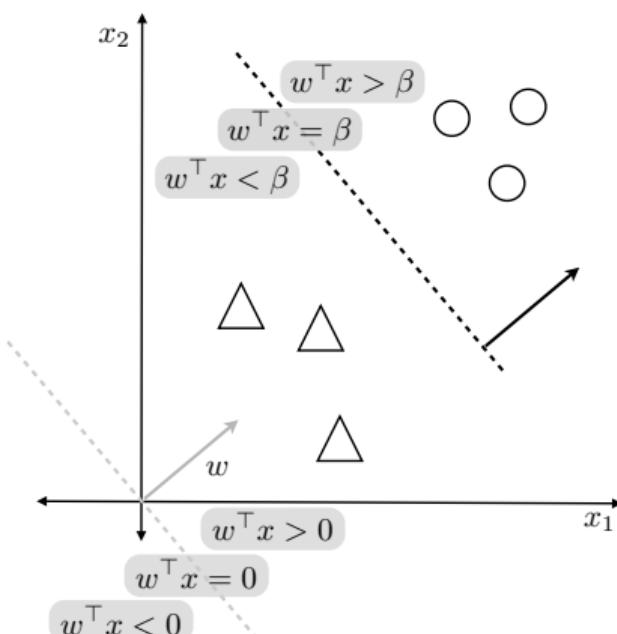
## Recap: Linear Classification



Linear decision boundary:

$$\mathbf{w}^\top \mathbf{x} - \beta = 0$$

## Recap: Nearest Centroid Classifier



Comparison of distance between data point  $\mathbf{x} \in \mathbb{R}^d$  to class means  $\bar{\mathbf{x}}_\Delta, \bar{\mathbf{x}}_o \in \mathbb{R}^d$  is equivalent to linear classification with

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

and

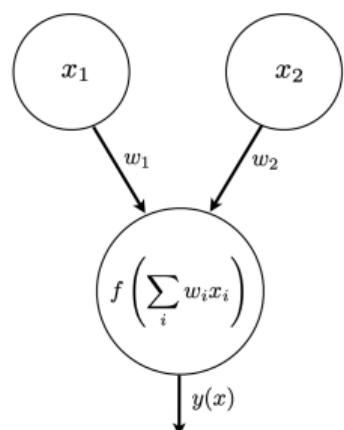
$$\beta = \frac{1}{2} \cdot \mathbf{w}^\top (\bar{\mathbf{x}}_o + \bar{\mathbf{x}}_\Delta)$$

Note notation change:  $\mathbf{w}_o = \bar{\mathbf{x}}_o$ ,  $\mathbf{w}_\Delta = \bar{\mathbf{x}}_\Delta$ .

$$= \frac{1}{2} \cdot (\bar{\mathbf{x}}_o - \bar{\mathbf{x}}_\Delta) (\bar{\mathbf{x}}_o + \bar{\mathbf{x}}_\Delta)$$

$$= \frac{1}{2} (\bar{\mathbf{x}}_o^\top \cdot \bar{\mathbf{x}}_o - \bar{\mathbf{x}}_\Delta^\top \cdot \bar{\mathbf{x}}_\Delta)$$

## Recap: Perceptron



Problem

Model

Loss function

Optimization

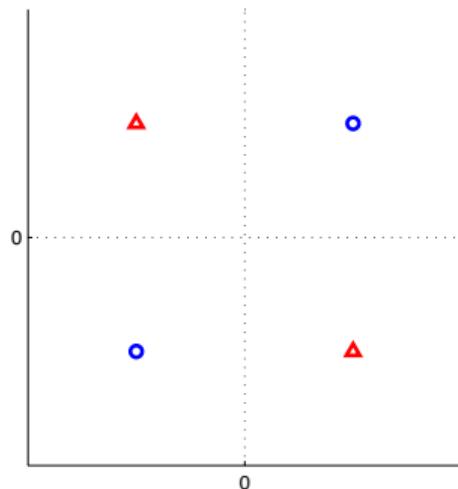
Classification

 $\hat{y} = f(\mathbf{w}^T \mathbf{x})$  $-\sum_{m \in \mathcal{M}} \mathbf{w}^T \mathbf{x}_m y_m$ 

stochastic gradient descent (SGD)

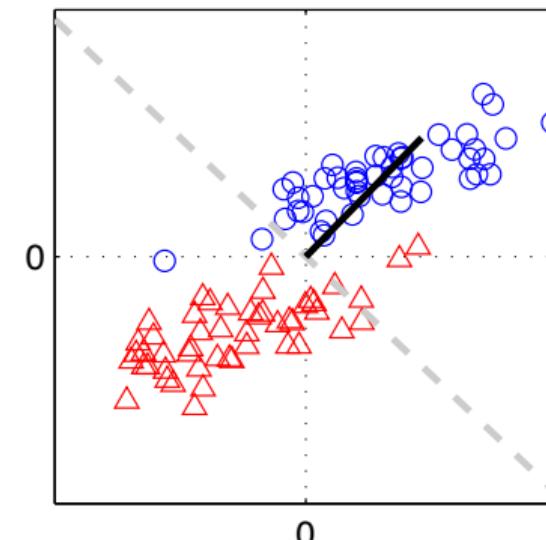
# Problems with Nearest Centroid Classification

Not linearly separable data



Solution  
Non-linear methods (later in this course)

Correlated data

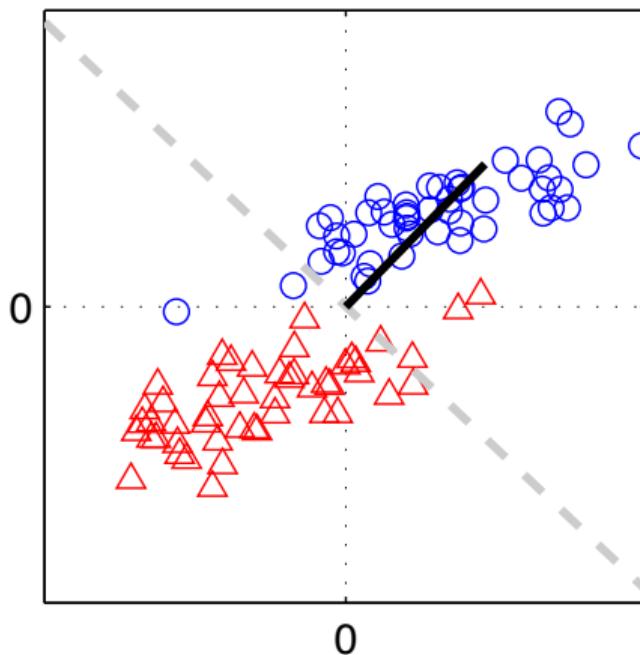


Solution  
(Fisher's) Linear Discriminant Analysis

## What is correlation?

Let's go through some definitions first

→ They will be useful later



## Random variables

Denote by  $\Omega$  the sample space, the set of all possible outcomes of an experiment.

A mapping  $X : \Omega \rightarrow \mathbb{R}$  which assigns a real value to every elementary event, is called a real-valued random variable.  
随机变量

用 $\Omega$ 表示样本空间，即实验的所有可能结果的集合。

一个映射  $X: \Omega \rightarrow \mathbb{R}$ , 将每个基本事件映射到一个实数值, 称为实值随机变量。

Example: coin toss  $\Rightarrow \Omega = \{\text{head}, \text{tail}\}$

$$X(\omega) = \begin{cases} 0, & \text{if } \omega = \text{tail} \\ 1, & \text{if } \omega = \text{head} \end{cases} \quad \text{for } \omega \in \Omega$$

- We use random variables to model the world
- In this course, we take a practical approach and introduce concepts when we need them



## Probabilities and expected values

If  $X$  is a **discrete random variable**, i.e. if  $X$  takes on only finitely<sup>1</sup> many values, we can assign probabilities  $p_i \in [0, 1]$  to the values  $x_i$  of  $X$ .

A probability of  $p_i$  means that out of very many trials, a fraction of  $p_i$  will have value  $x_i$ .

如果X是一个离散随机变量，即X只取有限多个值，我们可以为X的各个可能值 $x_i$ 分配概率 $p_i$ ，其中 $p_i \in [0, 1]$ 。

概率 $p_i$ 表示，在很多次试验中，会有 $p_i$ 的比例具有值 $x_i$ 。

The **expected value** of  $X$  is given by

$$\mathbb{E}[X] = \sum_i p_i x_i .$$

Example: coin toss with  $p_0 = p_1 = \frac{1}{2}$ , then

$$\mathbb{E}[X] = 0 \cdot p_0 + 1 \cdot p_1 = \frac{1}{2} .$$

---

<sup>1</sup>Strictly speaking, a discrete random variable takes finitely many or countably many values.

## Probability distributions and expected values

The probabilities of the values of a **continuous random variable** are described by a **probability density function**, a function  $p : X(\Omega) \rightarrow \mathbb{R}_+$  with  $\int_{X(\Omega)} p(x) dx = 1$ .

The probability of observing a value in  $[a, b] \subset \mathbb{R}$  is given by

$$\int_a^b p(x) dx .$$

概率分布和期望值

连续随机变量的值的概率由概率密度函数来描述，这是一个函数  $p: X(\Omega) \rightarrow \mathbb{R}_+$ ，其中  $\mathbb{R}_+$  表示正实数。观察到值在  $[a, b] \subset \mathbb{R}$  的概率可以由以下方式给出：【缺少具体公式，如果您需要具体公式的翻译，请提供公式内容。】

The expected value of  $X$  is given by

$$\mathbb{E}[X] = \int_{-\infty}^{\infty} x \cdot p(x) dx .$$

## 方差

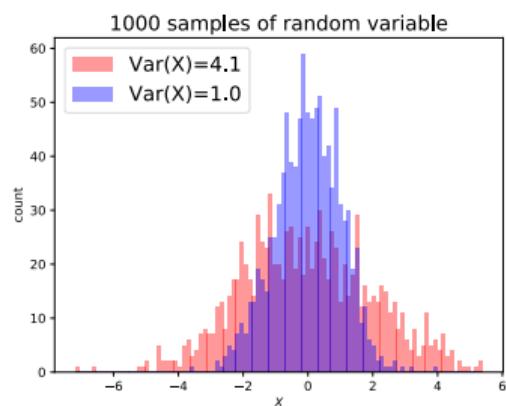
方差是随机变量X关于其均值的变异性的度量，通常表示为 $\text{Var}(X)$ ，计算方式为 $\text{Var}(X) = E[(X - E[X])^2]$ ，其中 $E[X]$ 表示X的期望值。

## Variance

协方差

measure of variability of  $X$  around its mean

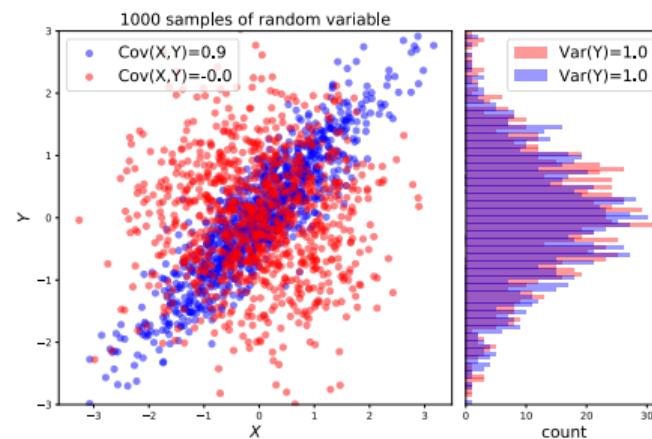
$$\text{Var}(X) = E[(X - E[X])^2]$$



## Covariance

measure of the joint variability of  $X$  and  $Y$

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



## 协方差 (Covariance) 和相关性 (Correlation)

## Covariance

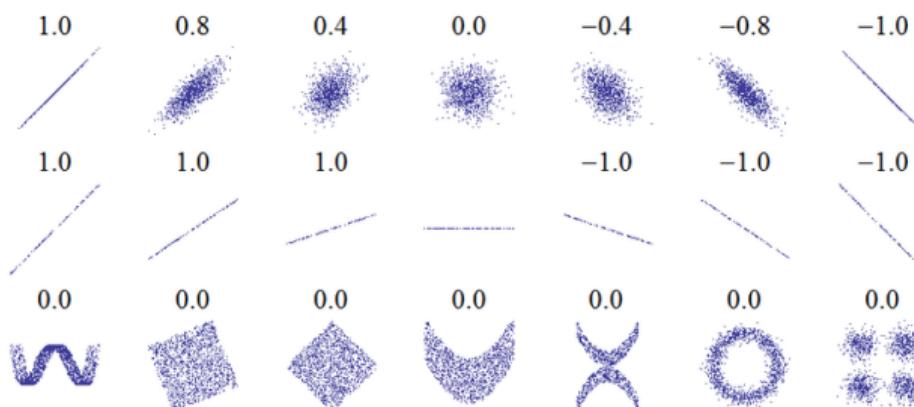
$$\text{Cov}(X, Y) := \mathbb{E}[(X - \mathbb{E}(X))(Y - \mathbb{E}(Y))]$$

$$\text{Cov}(X, Y) \leftarrow \sqrt{\text{Var}(X)} \sqrt{\text{Var}(Y)}$$

## Correlation

$$\text{Corr}(X, Y) := \frac{\text{Cov}(X, Y)}{\sqrt{\text{Var}(X)} \sqrt{\text{Var}(Y)}} \in [-1, 1].$$

normalized covariance

Indicate the strength of a **linear** relationship

## Correlation vs. dependence vs. causation

Consider two random variables  $X, Y$ .

- $X$  and  $Y$  are called **independent** if  $p(X, Y) = p(X) \cdot p(Y)$
- $X$  and  $Y$  are called **uncorrelated** if  $\text{Corr}(X, Y) = 0$

We might call  $X$  and  $Y$  **causally related** if  $X$  influences  $Y$  or vice versa.

### Note

- $X$  and  $Y$  independent implies  $X$  and  $Y$  uncorrelated
- $X$  and  $Y$  uncorrelated *does not* imply  $X$  and  $Y$  independent!  
(example  $Y = X^2$  on  $[-1, 1]$ )
- $X$  and  $Y$  dependent does not imply  $X$  and  $Y$  causally related

# Normal distribution

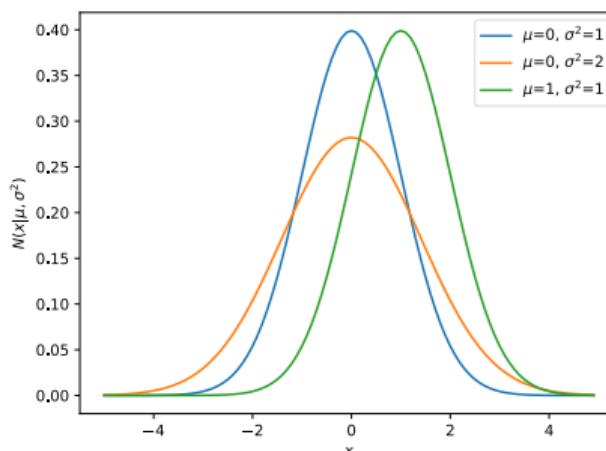
Parameters:

- Mean  $\mu \in \mathbb{R}$
- Variance  $\sigma^2 \in \mathbb{R}$

The probability density function

$$\mathcal{N}(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2} \frac{(x - \mu)^2}{\sigma^2}\right)$$

defines the **normal distribution** or **Gaussian distribution** with parameters  $\mu, \sigma^2$ .



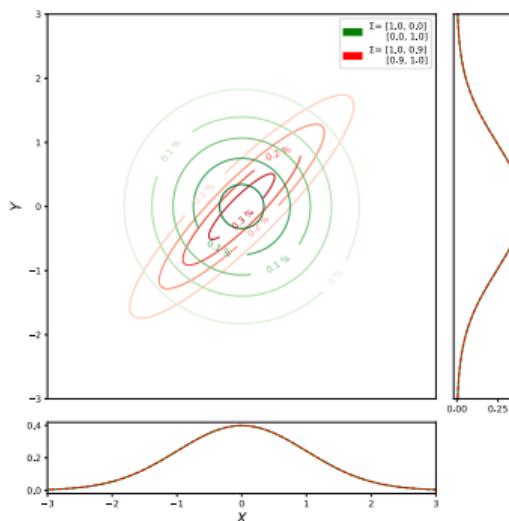
---

The parameters  $\mu$  and  $\sigma^2$  are called ‘mean’ and ‘variance’, because the mean  $\mathbb{E}[X]$  and variance  $\mathbb{E}[(X - \mathbb{E}[X])^2]$  of the distribution are  $\mu$  and  $\sigma^2$

# Multivariate normal distribution

For  $d$  dimensions:

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = (2\pi)^{-\frac{d}{2}} \det(\boldsymbol{\Sigma})^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})\right)$$



Parameters:

- Mean  $\boldsymbol{\mu} \in \mathbb{R}^d$
- Covariance matrix  $\boldsymbol{\Sigma} \in \mathbb{R}^{d \times d}$

## Estimating the covariance matrices

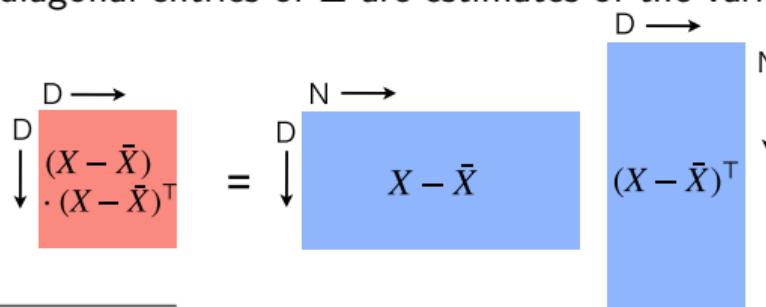
Given  $n$  data points  $\mathbf{x}_i \in \mathbb{R}^D$  in a data matrix  $X \in \mathbb{R}^{D \times n}$   
the empirical estimate of the **covariance matrix** is defined as

$$\hat{\Sigma} = \frac{1}{n} (X - \bar{X})(X - \bar{X})^\top,$$

where the estimate of the expected value is given by the mean

$$\bar{\mathbf{x}} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i, \quad \bar{X} = (\bar{\mathbf{x}}, \bar{\mathbf{x}}, \dots, \bar{\mathbf{x}}) \in \mathbb{R}^{D \times n}$$

The diagonal entries of  $\hat{\Sigma}$  are estimates of the variance.

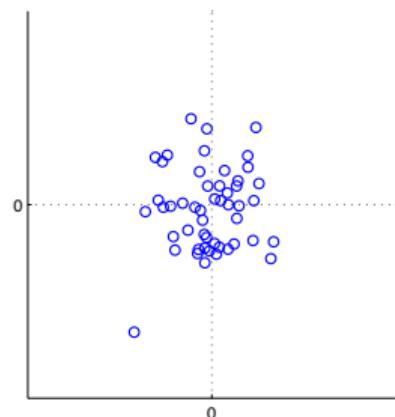


We call  $(X - \bar{X})(X - \bar{X})^\top$  the *empirical scatter matrix*

## Create correlated data from uncorrelated data

We can generate correlated data using a diagonal scaling matrix  $D$  and a rotation  $R$ . We assume centered data here (i.e.  $\bar{X} = 0$ ), so  $\hat{\Sigma} = \frac{1}{n}XX^\top$

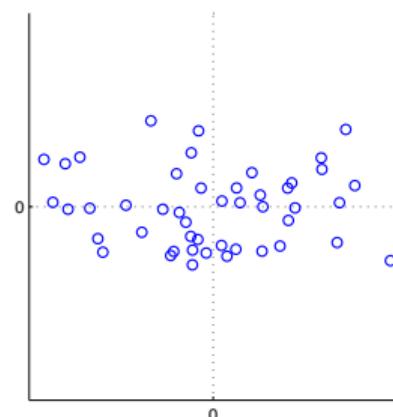
Uncorrelated



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

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

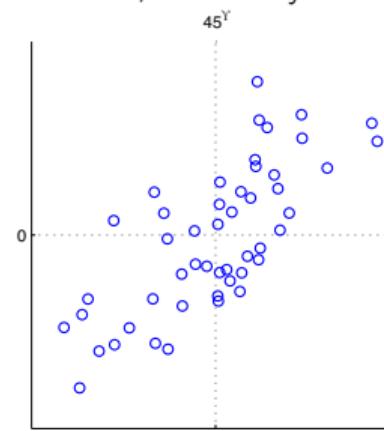
Uncorrelated, scaled



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

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

Scaled, rotated by 45°



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

$$\frac{1}{n}XX^\top = \begin{bmatrix} 5 & 4 \\ 4 & 5 \end{bmatrix}$$

## Ronald A. Fisher



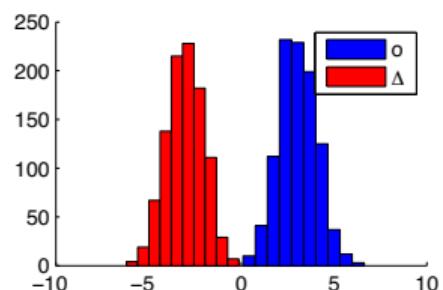
Ronald A. Fisher (1890 – 1962)

Founder of modern statistics  
Interested in Biology  
Suggested *Linear Discriminant Analysis* (LDA)  
Held some very problematic opinions

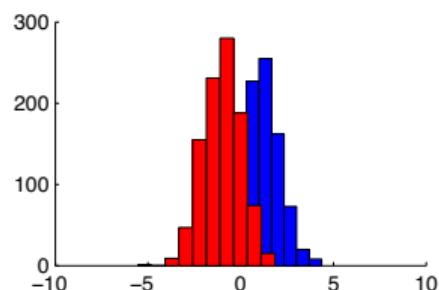
# The Fisher Criterion - measure for class separability

Consider one dimensional data and two classes

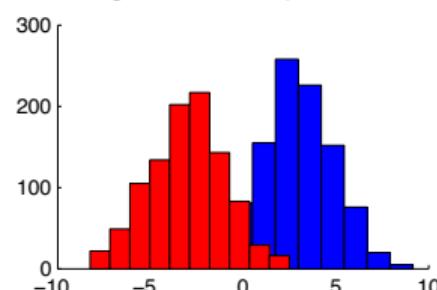
Good Class Separation



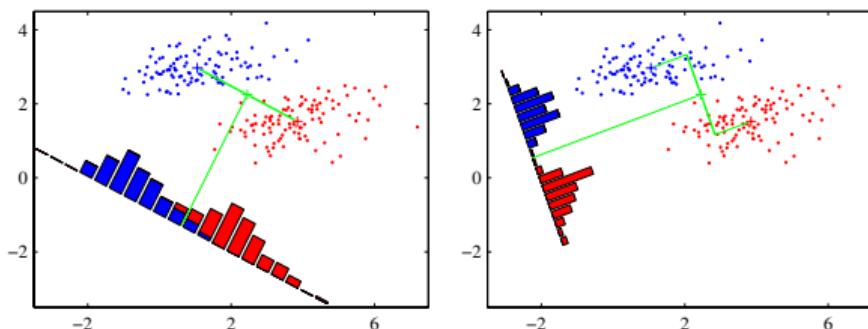
Bad Class Separation:  
Close means



Bad Class Separation:  
Large Variance per 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, and
- Minimizes variance in each class

Maximize the **Fisher criterion**:

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

where  $\mathbf{x}_{1o}, \dots, \mathbf{x}_{no} \in \mathbb{R}^d$  and

# Linear Discriminant Analysis

Rewrite Fisher criterion to separate out  $\mathbf{w}$ -dependence using

$$\bar{\mathbf{x}}_o := \frac{1}{n_o} \sum_{i=1}^{n_o} \mathbf{x}_{io} \quad \Rightarrow \quad \mu_o = \frac{1}{n_o} \sum_{i=1}^{n_o} \mathbf{w}^\top \mathbf{x}_{io} = \mathbf{w}^\top \bar{\mathbf{x}}_o, \quad \sigma_o = \frac{1}{n_o} \sum_{i=1}^{n_o} (\mathbf{w}^\top \mathbf{x}_{io} - \mu_o)^2 = \frac{1}{n_o} \sum_{i=1}^{n_o} (\mathbf{w}^\top (\mathbf{x}_{io} - \bar{\mathbf{x}}_o))^2$$

Hence,

$$(\mu_o - \mu_\Delta)^2 = (\mathbf{w}^\top (\bar{\mathbf{x}}_o - \bar{\mathbf{x}}_\Delta))^2 = \mathbf{w}^\top \underbrace{(\bar{\mathbf{x}}_o - \bar{\mathbf{x}}_\Delta)(\bar{\mathbf{x}}_o - \bar{\mathbf{x}}_\Delta)^\top}_{{S_B} - \text{"between class scatter"}} \mathbf{w}.$$

$$\begin{aligned} \sigma_o^2 + \sigma_\Delta^2 &= \frac{1}{n_o} \sum_{i=1}^{n_o} (\mathbf{w}^\top (\mathbf{x}_{io} - \bar{\mathbf{x}}_o))^2 + \frac{1}{n_\Delta} \sum_{j=1}^{n_\Delta} (\mathbf{w}^\top (\mathbf{x}_{j\Delta} - \bar{\mathbf{x}}_\Delta))^2 \\ &= \mathbf{w}^\top \underbrace{\left[ \frac{1}{n_o} \sum_{i=1}^{n_o} (\mathbf{x}_{io} - \bar{\mathbf{x}}_o)(\mathbf{x}_{io} - \bar{\mathbf{x}}_o)^\top + \frac{1}{n_\Delta} \sum_{j=1}^{n_\Delta} (\mathbf{x}_{j\Delta} - \bar{\mathbf{x}}_\Delta)(\mathbf{x}_{j\Delta} - \bar{\mathbf{x}}_\Delta)^\top \right]}_{S_W - \text{"within class scatter"}} \mathbf{w}. \end{aligned}$$

And therefore ( $\mathbf{w} \neq 0$ ),

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

## Linear Discriminant Analysis

The optimal weight vector  $\mathbf{w}$  is given by

$$\mathbf{w} = \underset{\mathbf{w}'}{\operatorname{argmax}} J(\mathbf{w}') = \underset{\mathbf{w}'}{\operatorname{argmax}} \frac{\mathbf{w}'^\top S_B \mathbf{w}'}{\mathbf{w}'^\top S_W \mathbf{w}'}$$

To optimize the Fisher criterion, we set its derivative (with respect to  $\mathbf{w}$ ) to 0

$$0 = \left. \frac{\partial}{\partial \mathbf{w}} J(\mathbf{w}) \right|_{\mathbf{w}} = \frac{(\mathbf{w}^\top S_W \mathbf{w}) S_B \mathbf{w} - (\mathbf{w}^\top S_B \mathbf{w}) S_W \mathbf{w}}{(\mathbf{w}^\top S_W \mathbf{w})^2}$$

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

$$S_W \mathbf{w} = S_B \mathbf{w} \underbrace{\frac{\mathbf{w}^\top S_W \mathbf{w}}{\mathbf{w}^\top S_B \mathbf{w}}}_{scalar \equiv \lambda}$$

## Linear Discriminant Analysis

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

Now we plug  $S_B = (\bar{x}_o - \bar{x}_\Delta)(\bar{x}_o - \bar{x}_\Delta)^\top$  in

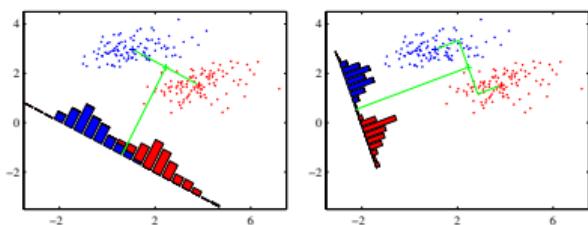
$$S_B \mathbf{w} = (\bar{x}_o - \bar{x}_\Delta) \underbrace{(\bar{x}_o - \bar{x}_\Delta)^\top \mathbf{w}}_{\text{scalar}}$$

finally, left multiplying with  $S_W^{-1}$  yields

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

( $\propto$  denotes proportionality, e.g.  $x \propto 2x$ ) 符号 " $\propto$ " 表示比例关系。

# Interim summary



## Goal

Find  $\mathbf{w} \in \mathbb{R}^d$  that

- maximizes mean class difference
- minimizes variance in each class

## Formalization

Maximize the **Fisher criterion**

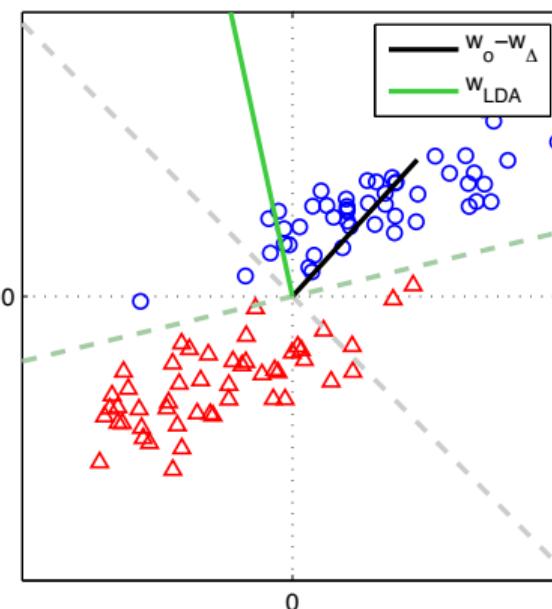
$$J(\mathbf{w}) = \frac{\text{between class variance}}{\text{within class variance}} = \frac{(\mu_o - \mu_\Delta)^2}{\sigma_o^2 + \sigma_\Delta^2}$$

## Solution

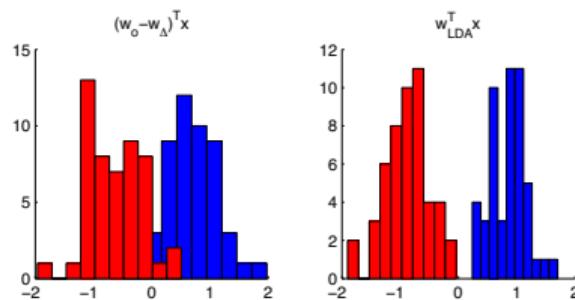
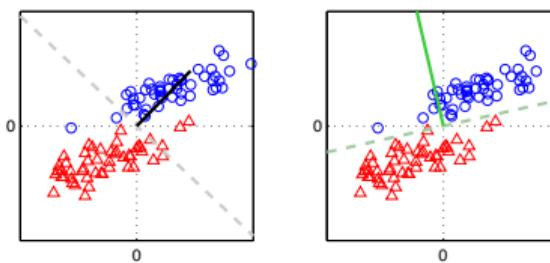
After some calculations...

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

# Linear Discriminant Analysis vs Nearest Centroid Classifier



# Linear Discriminant Analysis vs Nearest Centroid Classifier



If correlated data are the problem, why don't we decorrelate the data and then apply the nearest-centroid classifier?

如果数据之间存在相关性问题，为什么不对数据进行去相关化处理，然后再应用最近质心分类器呢？

如何去相关化数据?

去相关化是指将数据转换成一个对角线形式的经验协方差矩阵  $\Sigma$ 。

白化则将数据转换成一个单位的  $\Sigma$ :

## How can we decorrelate the data?

- *Decorrelating* refers to transforming to a diagonal empirical covariance matrix  $\hat{\Sigma}$
- *Whitening* transforms to a unit  $\hat{\Sigma}$ :
  - 1 For a data matrix  $X \in \mathbb{R}^{D \times n}$ , calculate  $\hat{\Sigma} = \frac{1}{n}(X - \bar{X})(X - \bar{X})^T$
  - 2 Calculate eigenvalue decomposition  $U\Lambda U^T = \hat{\Sigma}$  with  $\Lambda$  diagonal
  - 3 Transform  $X$ :  $\tilde{X} = \Lambda^{-1/2}U^T X$

## New Covariance

$$\begin{aligned}\tilde{\Sigma} &= \frac{1}{n}(\tilde{X} - \bar{\tilde{X}})(\tilde{X} - \bar{\tilde{X}})^T \\ &= \Lambda^{-1/2} U^T \underbrace{\frac{1}{n}(X - \bar{X})(X - \bar{X})^T}_{\hat{\Sigma}=U\Lambda U^T} U \Lambda^{-1/2} \\ &= \Lambda^{-1/2} U^T U \Lambda U^T U \Lambda^{-1/2} \\ &= I\end{aligned}$$

There is more than one way of whitening (we can multiply  $\tilde{X}$  with any orthogonal matrix  $OO^T = I$ )

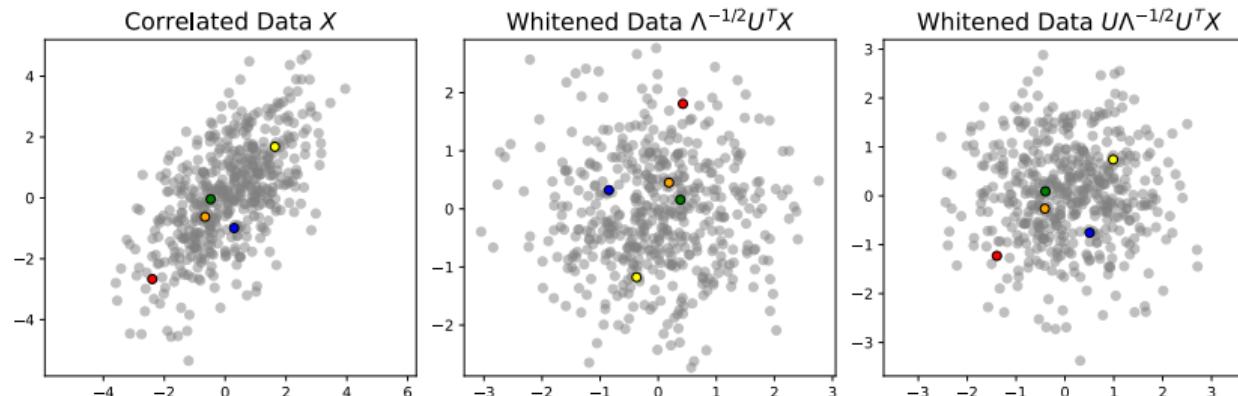
# Whitening

Transforms data to data with covariance matrix that is the identity.

→ Data are decorrelated after whitening

Often used as part of preprocessing

Leads to more numeric stability



# Linear Discriminant Analysis 线性判别分析

For centered data, we have

$$S = \frac{1}{n_\Delta + n_o} XX^T = S_W + \frac{n_\Delta n_o}{n_\Delta + n_o} S_B$$

Compare Subsection 4.1.5 in PRML<sup>2</sup> and Exercise 4.6 in PRML, a solution of the exercise is available here  
<https://github.com/zhengqigao/PRML-Solution-Manual>.

Then, for the LDA weight vector  $\mathbf{w}$ :

$$S_W \mathbf{w} \propto S_B \mathbf{w}$$

$$(S - \frac{n_\Delta n_o}{n_\Delta + n_o} S_B) \mathbf{w} \propto S_B \mathbf{w}$$

$$S \mathbf{w} \propto S_B \mathbf{w} \propto \bar{\mathbf{x}}_o - \bar{\mathbf{x}}_\Delta$$

$$\mathbf{w} \propto S^{-1}(\bar{\mathbf{x}}_o - \bar{\mathbf{x}}_\Delta)$$

---

<sup>2</sup>C. M. Bishop, Pattern Recognition and Machine Learning, freely available here.

# Linear Discriminant Analysis

The predictions of LDA then are:

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

$$\mathbf{w}^T \mathbf{x} \propto (\bar{\mathbf{x}}_o - \bar{\mathbf{x}}_\Delta)^T S^{-1} \mathbf{x} \propto \underbrace{(\bar{\mathbf{x}}_o - \bar{\mathbf{x}}_\Delta)^T U \Lambda^{-1/2} \Lambda^{-1/2} U^T \mathbf{x}}_{\text{mean class difference of whitened } X}$$

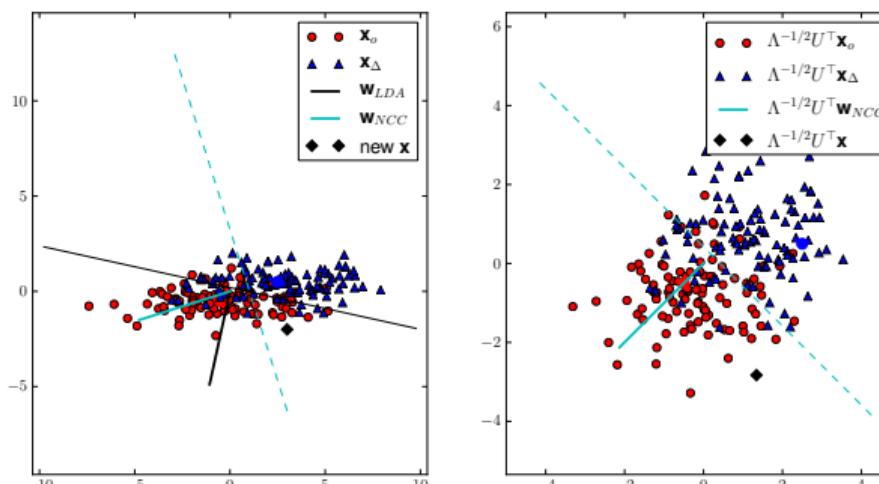
where  $S = U \Lambda U^T$  is the eigenvalue decomposition of  $S$

## Linear Discriminant Analysis

Alternative view: For centered data, LDA first whitens the data followed by nearest centroid classification:

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

where  $S = U \Lambda U^T$  is the eigenvalue decomposition of  $S$

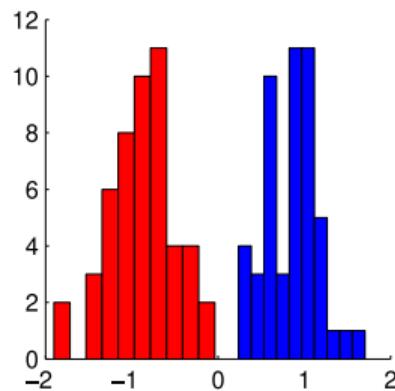


## Discriminative and Generative Model

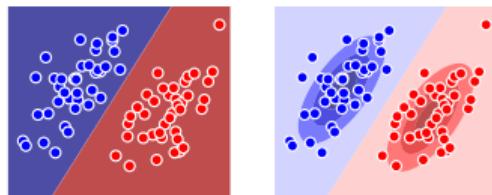
So far:

Find one-dimensional projection via  $w$  which best separates the two classes.

How can we build a discriminator, i.e. find a bias?



Generative approach:  
Let's model how data was generated



Can e.g. use center between projected means, i.e.  
 $\beta = \frac{1}{2}(\mu_o + \mu_\Delta)$ .

## Probabilistic modelling

Decision theory: The optimal classifier is Bayes classifier

For a new data point  $\mathbf{x} \in \mathbb{R}^d$

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

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

$$p(\Delta|\mathbf{x}) = \frac{p(\Delta)p(\mathbf{x}|\Delta)}{p(\mathbf{x})}$$

For the decision,  $p(\mathbf{x})$  is irrelevant:

$$p(\Delta|\mathbf{x}) > p(o|\mathbf{x}) \iff p(\Delta)p(\mathbf{x}|\Delta) > p(o)p(\mathbf{x}|o).$$

## Probabilistic modelling

The class probabilities  $p(\Delta)$ ,  $p(o)$  can be estimated using

$$p(\Delta) \approx \frac{n_\Delta}{n_\Delta + n_o} \quad \text{and similarly for } o.$$

Estimating  $p(x|\Delta)$  is difficult:

→ if each dimension of  $x$  can take 2 values →  $2^d$  possible values.

One solution:

Choose distributions for  $p(x|\Delta)$ ,  $p(x|o)$  that are easy to deal with.

→ Most popular: The Gaussian (or normal) distribution

$$x \in \mathbb{R}^d \text{ in class } \Delta \sim \mathcal{N}(\bar{x}_\Delta, S_\Delta) = \frac{1}{(2\pi)^{\frac{d}{2}} \sqrt{\det(S_\Delta)}} e^{-\frac{1}{2}(x-\bar{x}_\Delta)^\top S_\Delta^{-1}(x-\bar{x}_\Delta)}$$

and similarly for class  $o$ .

## Linear discriminant - a probabilistic view

If we use equal covariance in each class,  $\bar{S} = \frac{1}{n_\Delta + n_o} (n_\Delta S_\Delta + n_o S_o)$ , the classification boundary is linear and given by

$$\begin{aligned}\mathbf{w} &= \bar{S}^{-1}(\bar{\mathbf{x}}_o - \bar{\mathbf{x}}_\Delta) \\ \beta &= \frac{1}{2} \mathbf{w}^T (\bar{\mathbf{x}}_o + \bar{\mathbf{x}}_\Delta) + \underbrace{\log \frac{p(o)}{p(\Delta)}}_{\text{vanishes for } p(o)=p(\Delta)} \\ &= \frac{1}{2} (\mu_o + \mu_\Delta) + \log \frac{p(o)}{p(\Delta)}\end{aligned}$$

From Fisher criterion, we got

$$\mathbf{w} \propto S_W^{-1}(\bar{\mathbf{x}}_o - \bar{\mathbf{x}}_\Delta) \quad \text{with} \quad S_W = S_\Delta + S_o \quad \text{and (e.g.)} \quad \beta = \frac{1}{2} (\mu_o + \mu_\Delta)$$

$\Rightarrow$  Same as above if  $n_\Delta = n_o$

## LDA summary

Problem    Classification

Model     $y = \text{sign}(\mathbf{w}^T \mathbf{x} - \beta)$

Error function     $\underset{\mathbf{w}}{\text{argmax}} \frac{\mathbf{w}^T S_B \mathbf{w}}{\mathbf{w}^T S_W \mathbf{w}}$

Optimization    Closed form

# LDA algorithm

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

**Input:** Data  $\{(x_1, y_1), \dots, (x_n, y_n)\}, x_i \in \mathbb{R}^d, y_i \in \{-1, +1\}$ ,

Compute class mean vectors

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

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

Compute averaged covariance matrix

$$\bar{S} = 1/(n_+ + n_-) \left[ \sum_{i \in \mathcal{Y}_-} (\mathbf{x}_i - \bar{\mathbf{x}}_-)(\mathbf{x}_i - \bar{\mathbf{x}}_-)^\top + \sum_{j \in \mathcal{Y}_+} (\mathbf{x}_j - \bar{\mathbf{x}}_+)(\mathbf{x}_j - \bar{\mathbf{x}}_+)^{\top} \right]$$

Compute normal vector  $\mathbf{w}$

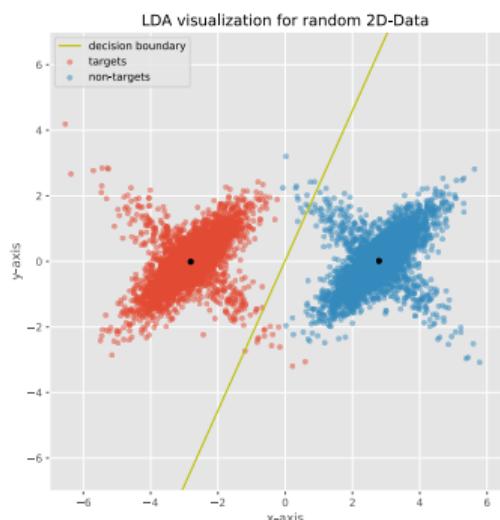
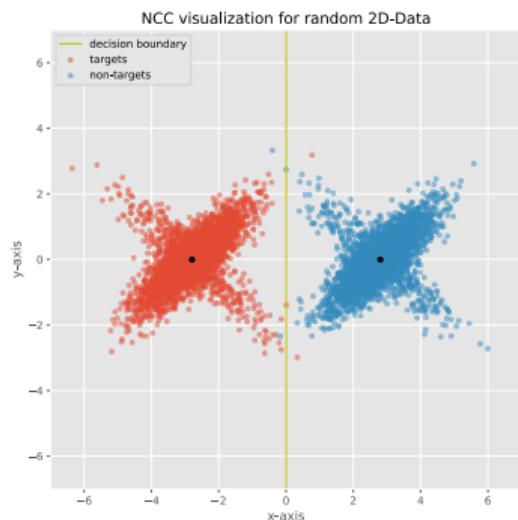
$$\mathbf{w} = \bar{S}^{-1}(\bar{\mathbf{x}}_+ - \bar{\mathbf{x}}_-)$$

Compute threshold

$$\beta = 1/2 \mathbf{w}^\top (\bar{\mathbf{x}}_+ + \bar{\mathbf{x}}_-) + \log(n_-/n_+)$$

**Output:**  $\mathbf{w}, \beta$

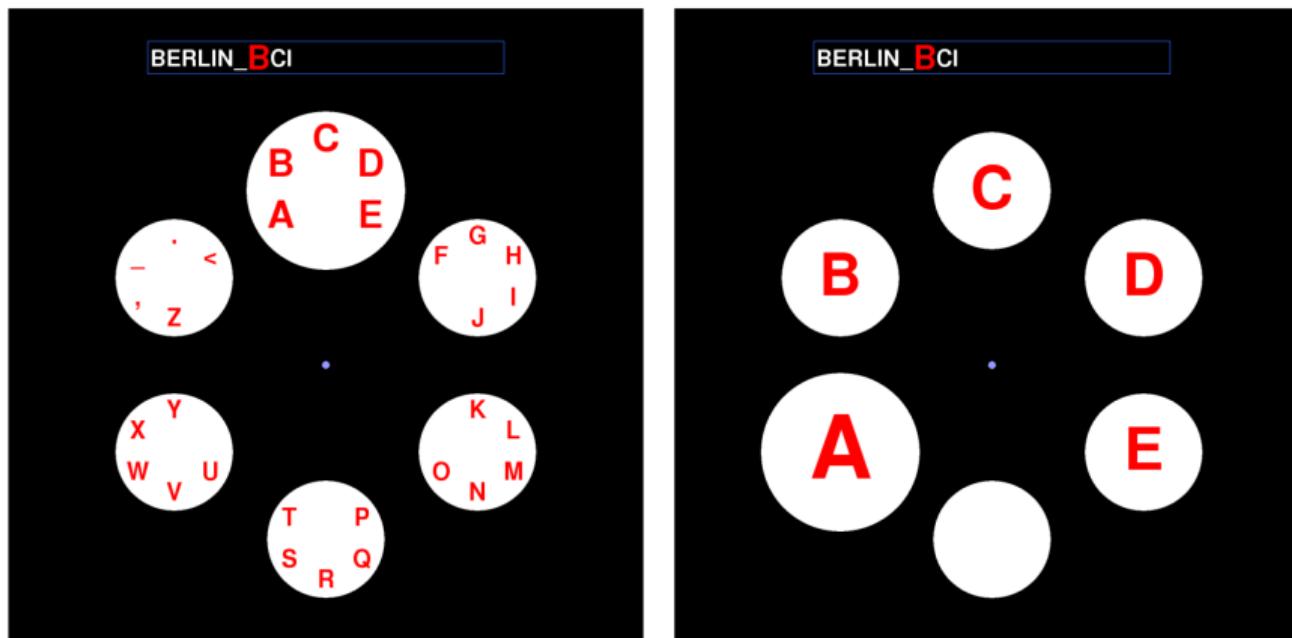
# Is LDA always better than NCC?



⇒ No, only if our assumption of equal covariance and Normal distribution for each class holds

# Berlin Brain-Computer-Interface (BBCI)

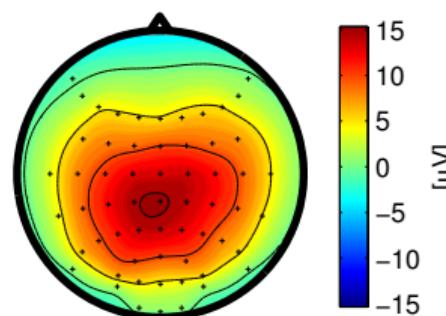
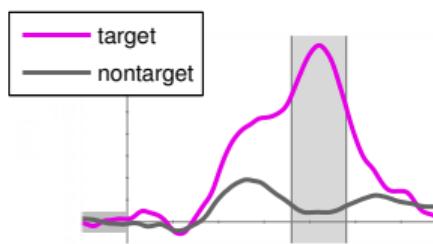
Hex-o-spell: Writing with thoughts



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

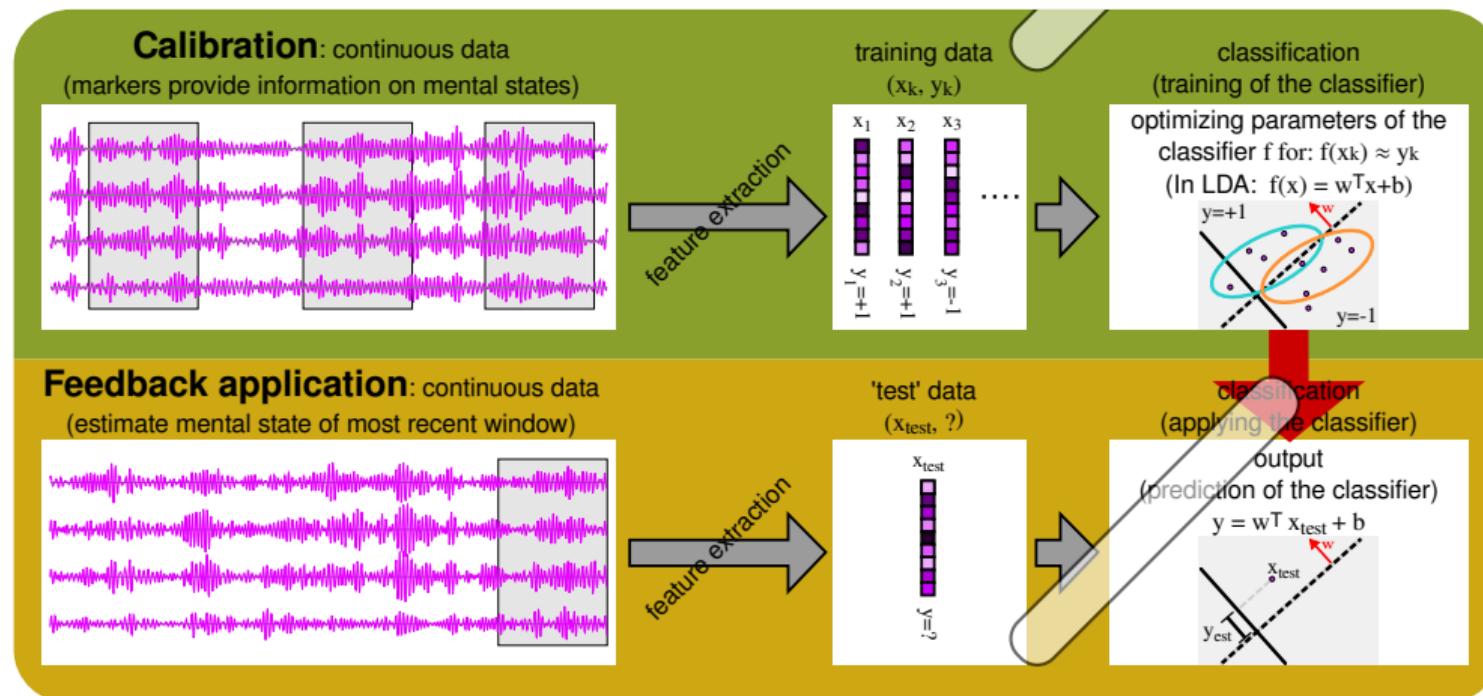
## BCI based on event-related potentials (ERPs)

- User concentrates on a symbol (the “target”)
- The six circles are intensified randomly
- Intensified targets elicit ERPs that differ from non-targets
- Training data is collected and an LDA classifier is trained
- The trained classifier can now be used for spelling



This Video explains the data gathering [00:43 - 3:05]

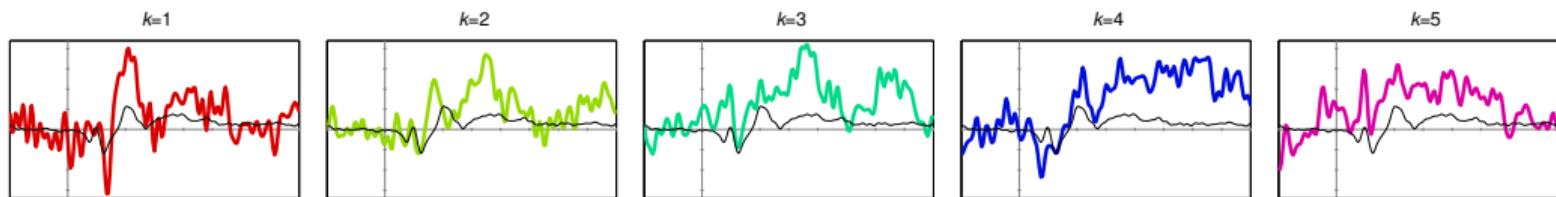
# BCI with ML: calibration and feedback



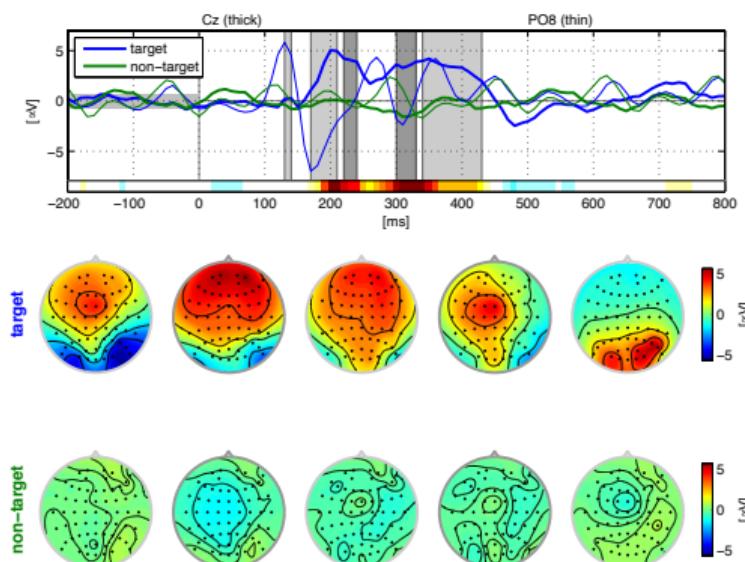
## Illustration: single trials and ERPs

### Continuous Signal (with markers)

### Segments (epochs) around stimulus markers



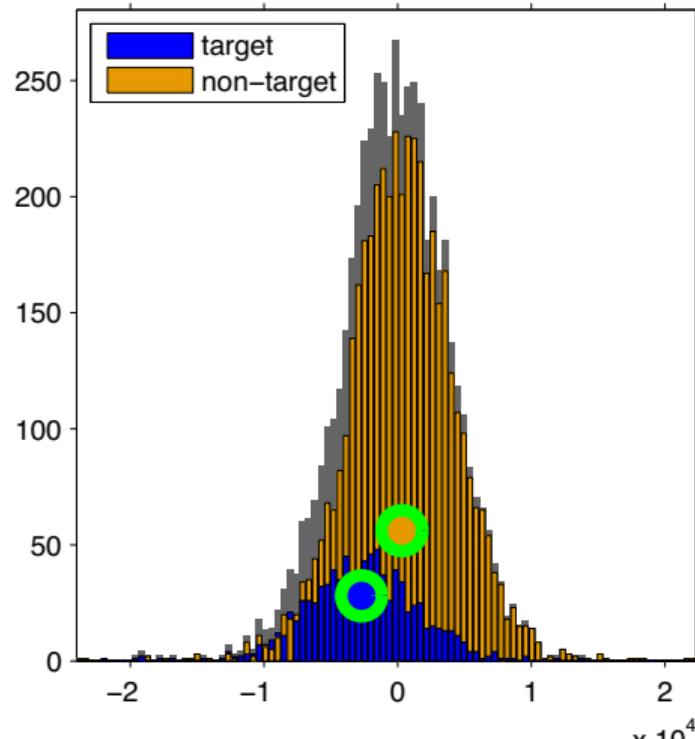
# Scalp potentials in response to targets/non-targets



# Berlin Brain-Computer-Interface

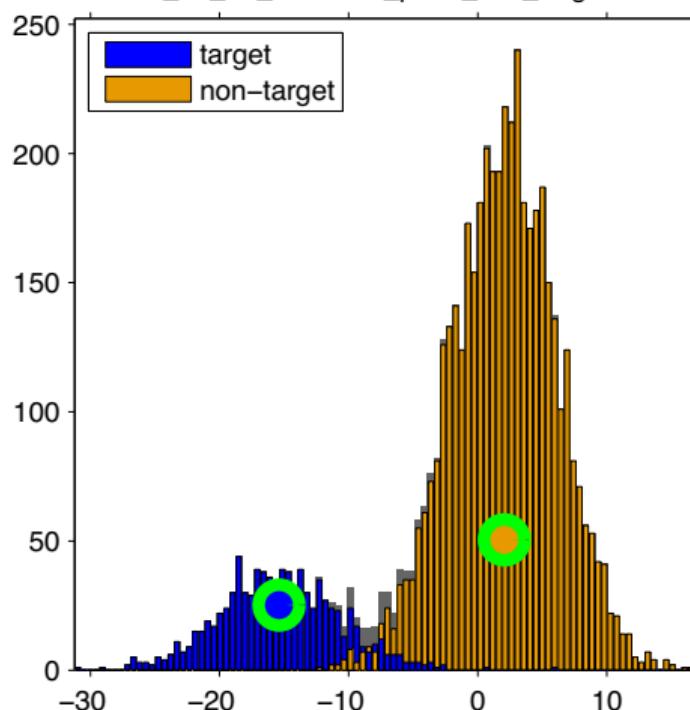
## Centroid Classification

VPsah\_09\_03\_16/visual\_p300\_hex\_targetVPsah



## Fisher's LDA

VPsah\_09\_03\_16/visual\_p300\_hex\_targetVPsah



## How can we properly evaluate a model?

If we use the following dataset  $X \in \mathbb{R}^{d \times n}$ :

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{0}, I)$$

$$p(y = +1 | \mathbf{x}) = 0.5$$

with  $n_{train} = 100$ ,  $d = 300$ .

We get the following accuracies

|       | Perceptron | NCC |
|-------|------------|-----|
| train | 100%       | 50% |
| test  | 50%        | 50% |

### Overfitting

The production of an analysis which corresponds too closely or exactly to a particular set of data, and may therefore fail to fit additional data or predict future observations reliably.

<https://www.lexico.com/definition/overfitting>

# Generalization and model evaluation

## Generalization

Generalization is the correct categorization/prediction of new (unseen) data

How can we estimate generalization performance?

- Train model and choose parameters on main part of data
- Test model on other part of data, *that was not seen during training*, to estimate overall performance

# Summary

## Correlation...

- ... is a measure of *linear relationship* between random variables
- ... between features can affect classification accuracy

## Linear Discriminant Analysis (LDA)

- LDA maximizes *between class variance* while minimizing *within class variance*
- For centered data, LDA is a NCC on whitened data
- If both classes follow a Gaussian with equal class covariances, then LDA is the optimal classifier

## Model evaluation

- Only looking at performance on *training set* will give us an overly optimistic estimate of performance (*overfitting*)
- We want our model to *generalize well*