Linear classification

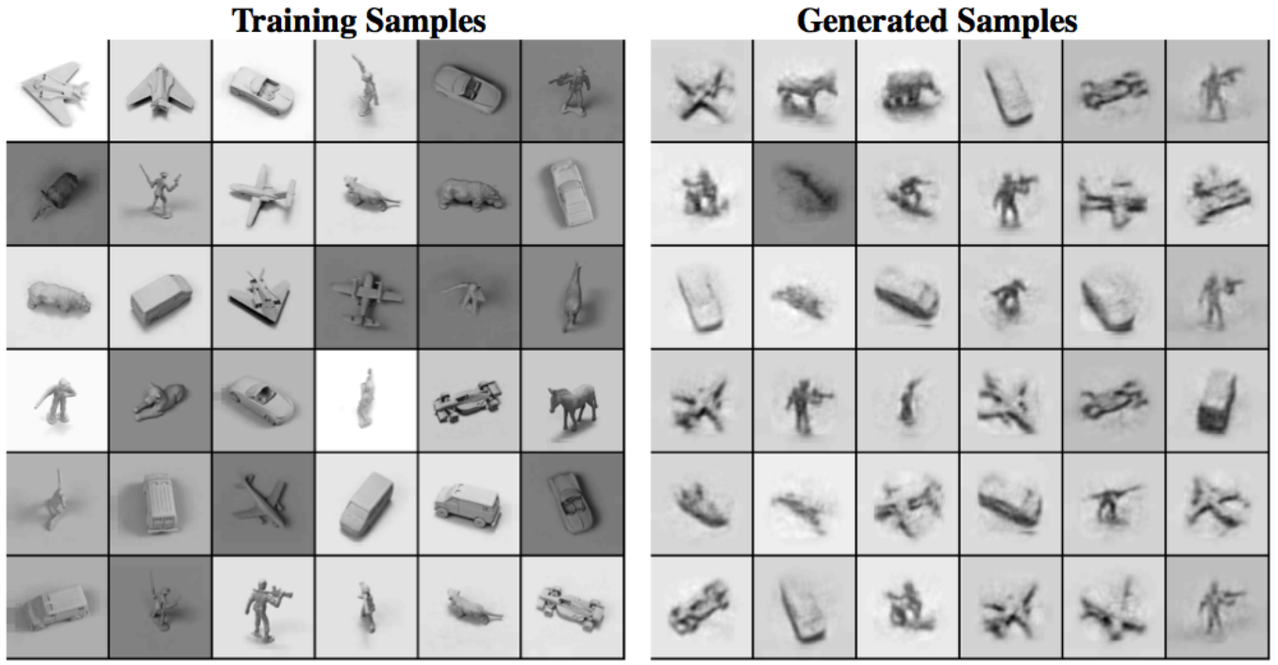
Machine Learning Group, TU Berlin

Outline

- Generative and discriminative models
- Perceptron
- Nearest Centroid Classifier
- Fisher’s Linear Discriminant
- Application in Brain-computer interfaces
- Component analysis: spatio-spectral decomposition
- Interpretation of linear models
- Non-linear models
- Correlation

Generative models

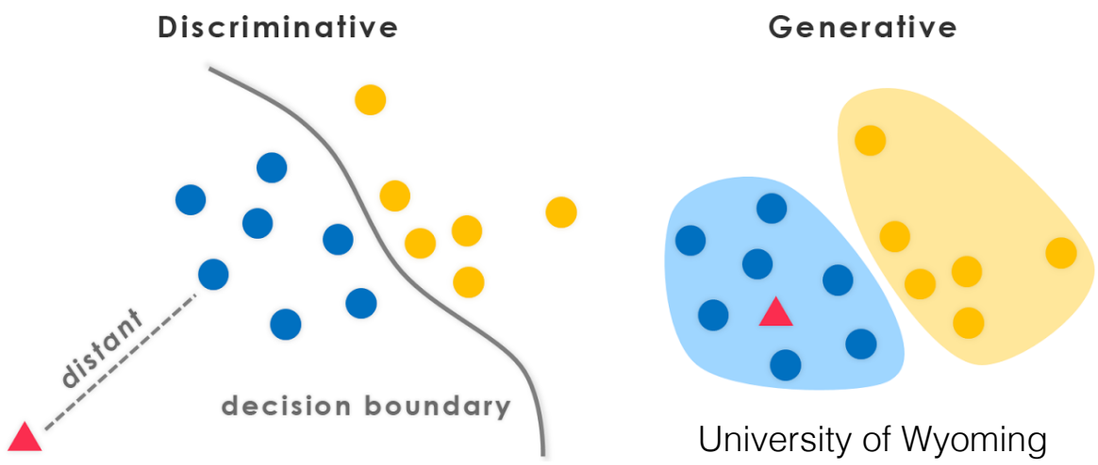
- Jointly model $p(\mathbf{x}, y)$
- Are able to generate new data samples $(\mathbf{x}_*, y_*)$ from $p(\mathbf{x}, y)$
- Example: Bayes-optimal classification $p(\omega_i, \mathbf{x}) = p(\mathbf{x}|\omega_i)p(\omega_i)$
- Helpful for interpretation/to assess what model has learned



Roger Grosse, metacademy

Discriminative models

- In many cases it is not necessary to know the data-generating distributions $p(\mathbf{x}|\omega_i)$
- All we care about is the $p(\omega_i|\mathbf{x}) = \frac{p(\mathbf{x}|\omega_i)p(\omega_i)}{\sum_j p(\mathbf{x}|\omega_j)p(\omega_j)}$
- Idea: try to estimate $p(\omega_i|\mathbf{x})$ directly without using $p(\mathbf{x}|\omega_i)$
- Simplifies learning task

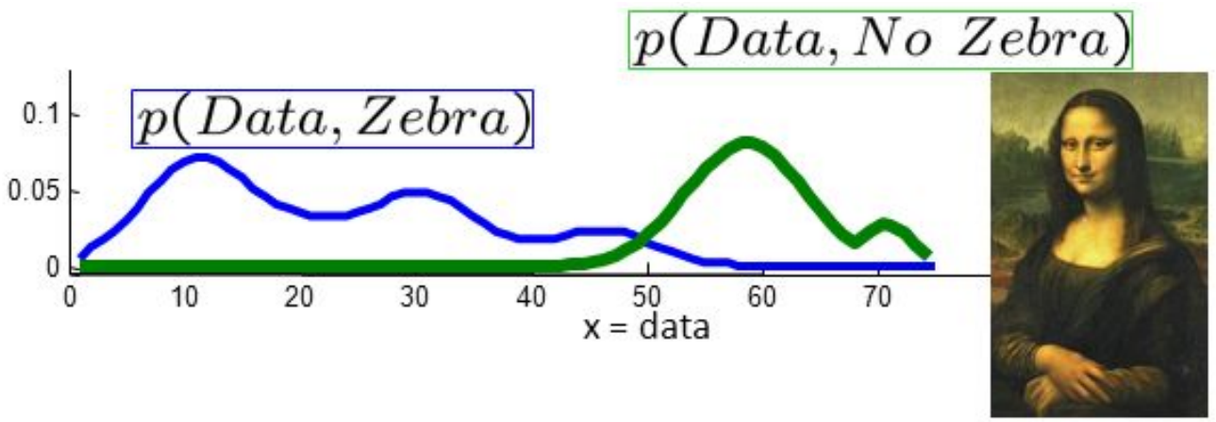


University of Wyoming

Discriminative vs. generative

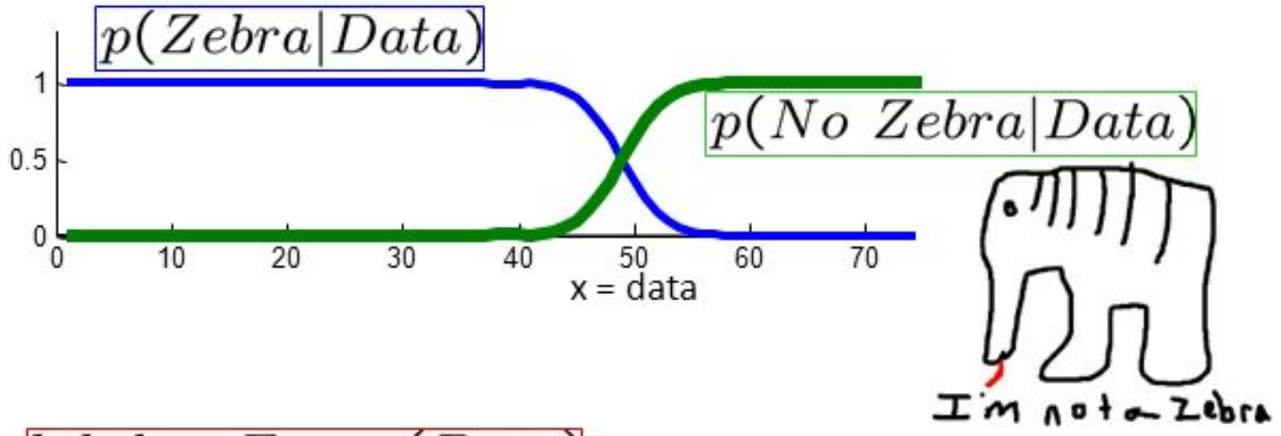
- Generative model

(*The artist*)

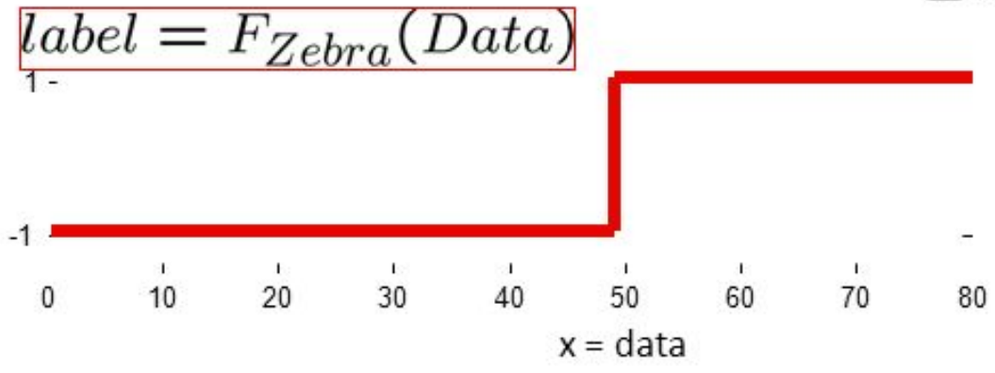


- Discriminative model

(The lousy painter)



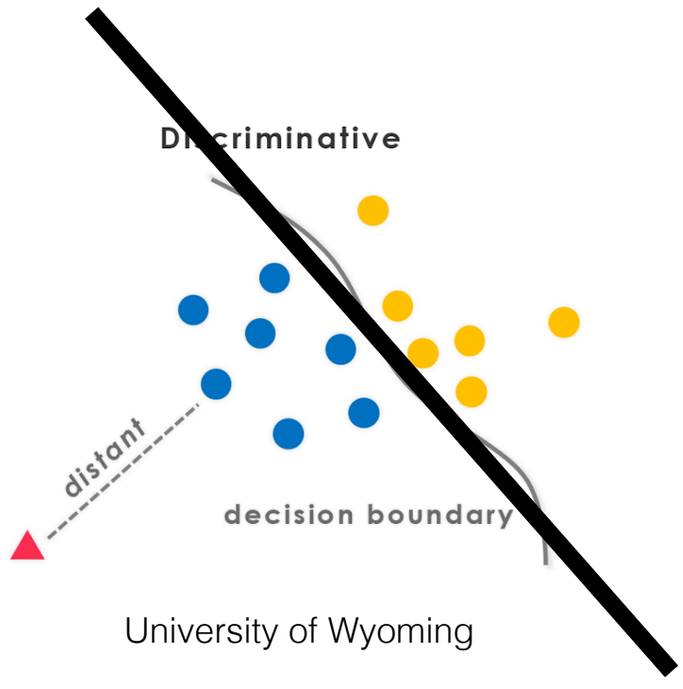
- Classification function



Luke Zettlemoyer, Carlos Guestrin, and Ben Taskar

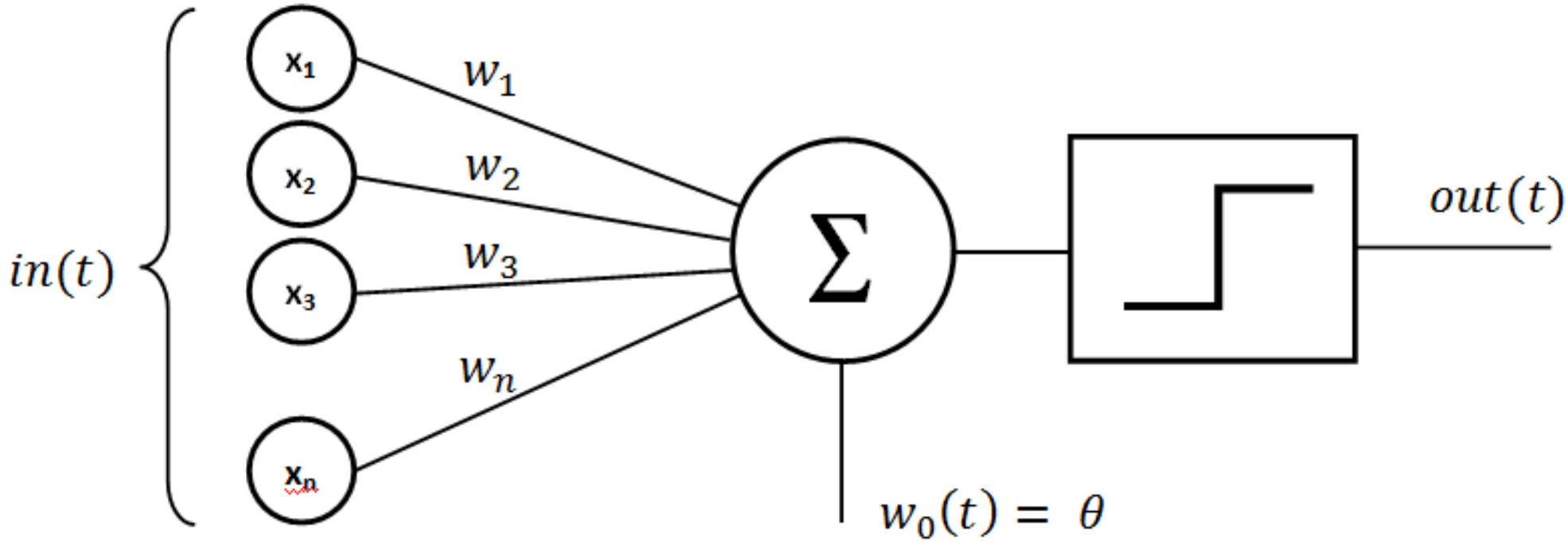
Linear classification

- $p(\omega_i|\mathbf{x})$ and $p(\mathbf{x}|\omega_i)$ and $p(\omega_i, \mathbf{x})$ are *d+1* dimensional objects
- In the Gaussian case: $\mathcal{O}(d^2)$ parameters
- The shape of these distributions determines the shape of the discriminant function (e.g., quadratic, linear)
- Why not directly constrain shape of discriminant?
- E.g., linear: $g(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$, *d+1* parameters



University of Wyoming

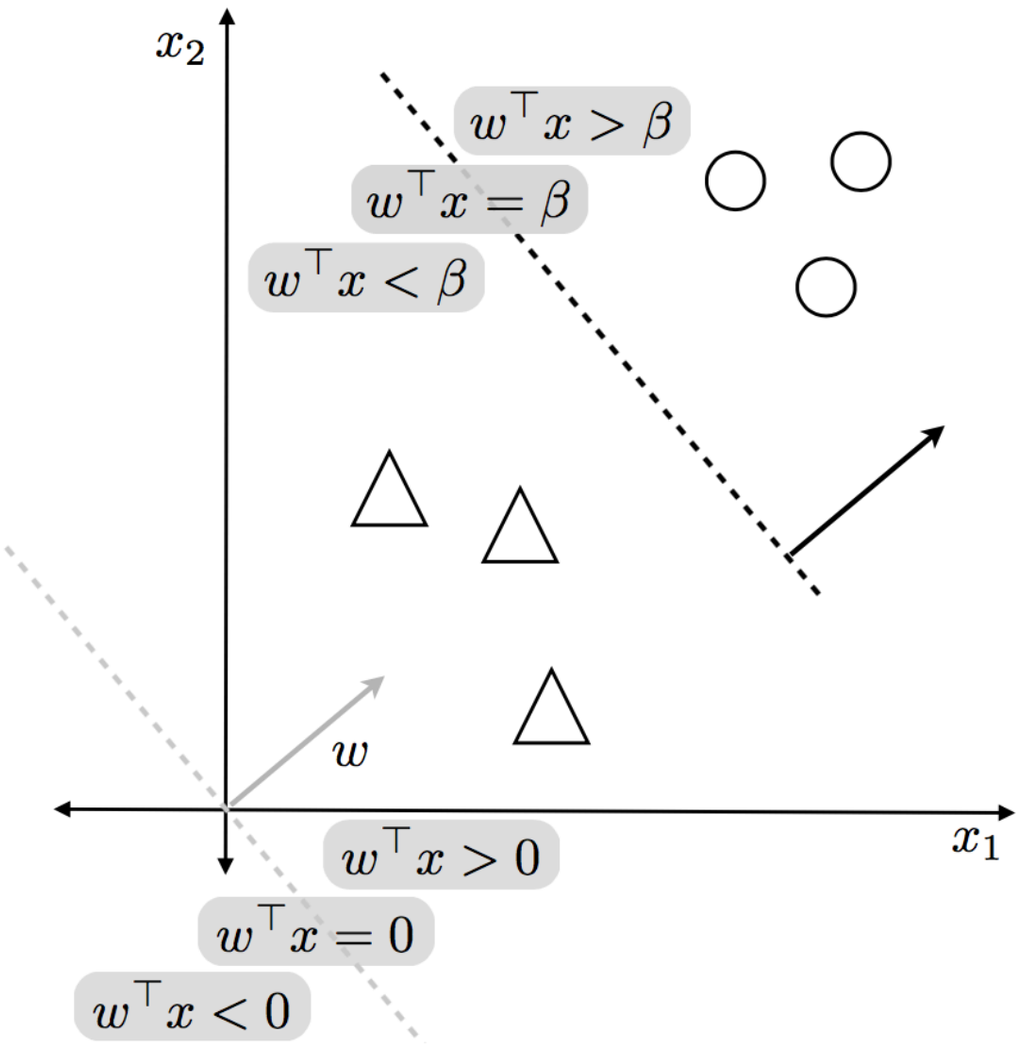
Perceptron



$$g(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$$

Simplest biologically inspired neural network (Rosenblatt, 1958)

Perceptron



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

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

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

Perceptron: algorithm

Goal: minimize error on misclassified data $\mathcal{E}(\mathbf{w}) = -\sum_{m \in \mathcal{M}} \mathbf{w}^\top \mathbf{x}_m y_m$

Algorithm: stochastic gradient descent

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

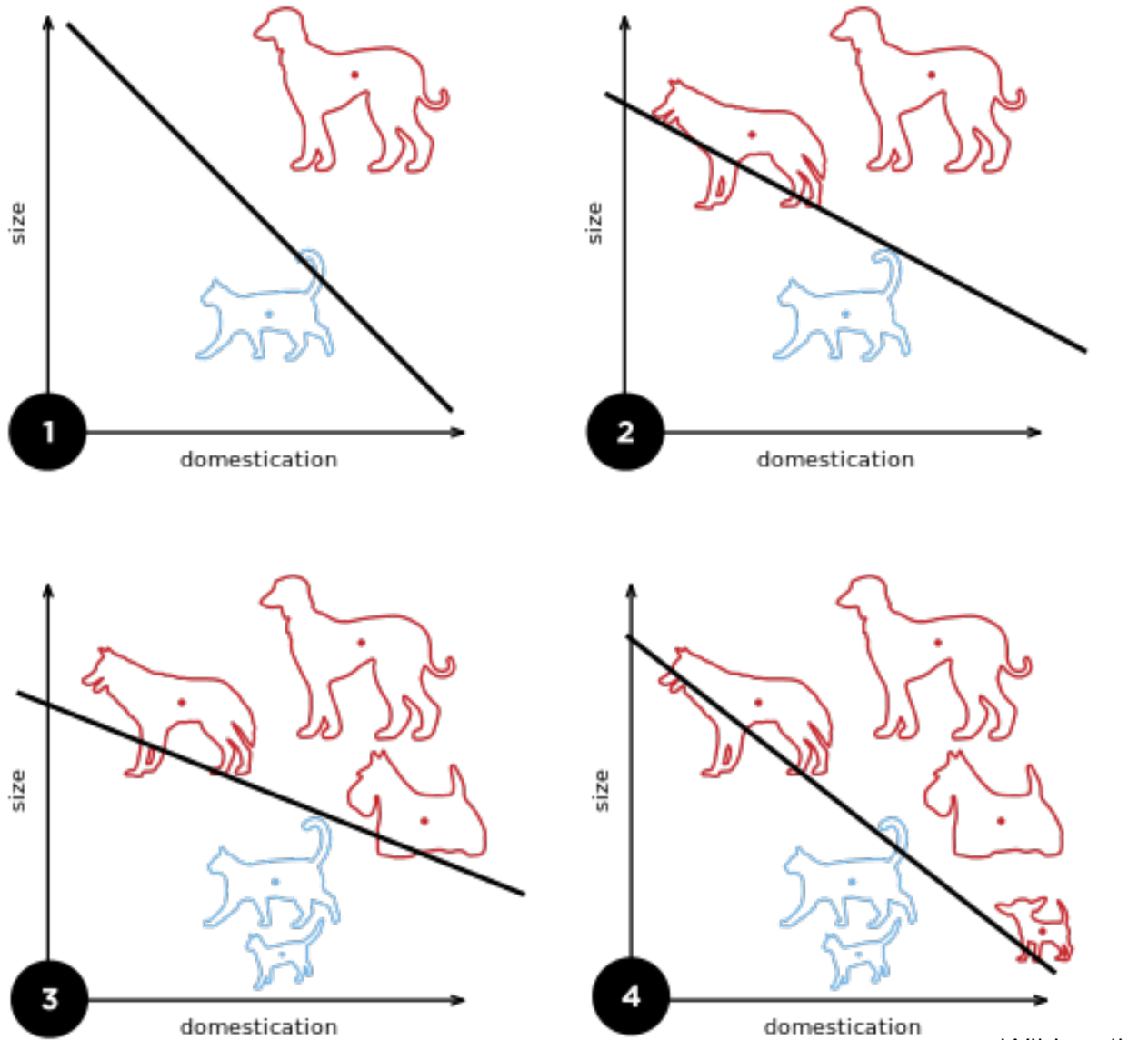
Input: Data $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}, \mathbf{x}_i \in \mathbb{R}^D,\ y_i \in \{-1, +1\}$,
learning rate η,
iterations N_{it}

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: $\mathbf{w}$

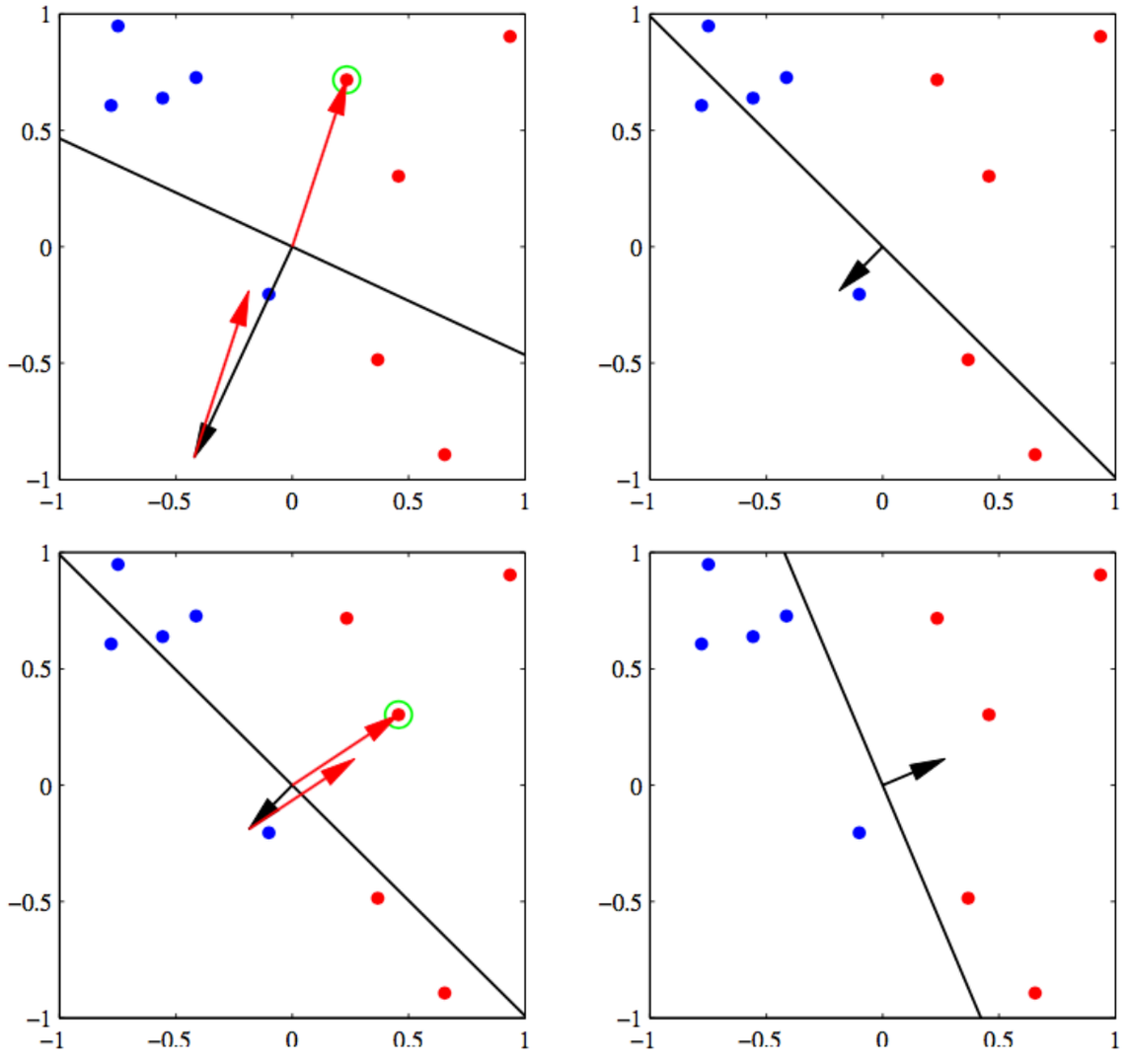
Perceptron: algorithm



Wikipedia

Perceptron: algorithm

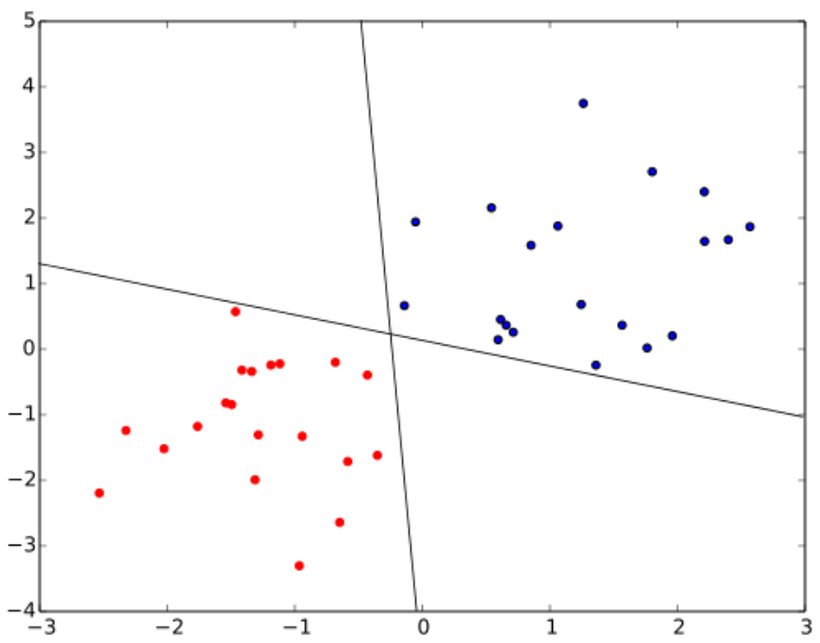
$$\mathbf{w}^{\text{new}} \leftarrow \mathbf{w}^{\text{old}} + \eta \mathbf{x}_m y_m$$



Christopher Bishop

Perceptron: limitations

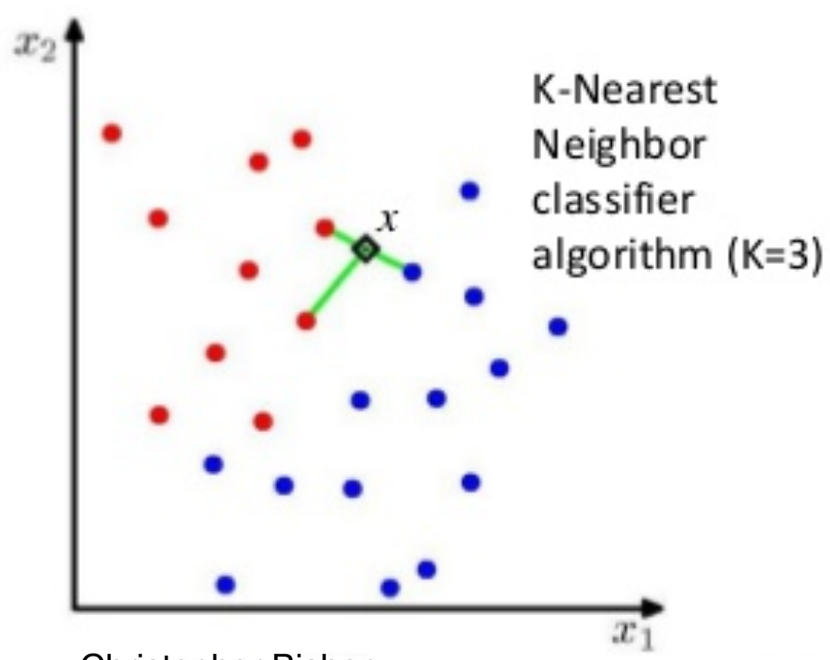
- Converges only for linearly separable data
- In separable case: many possible solutions



k-Nearest-neighbor classifier

- Psychologists: we learn from prototypes

→ Classify based on distance to nearest neighbors

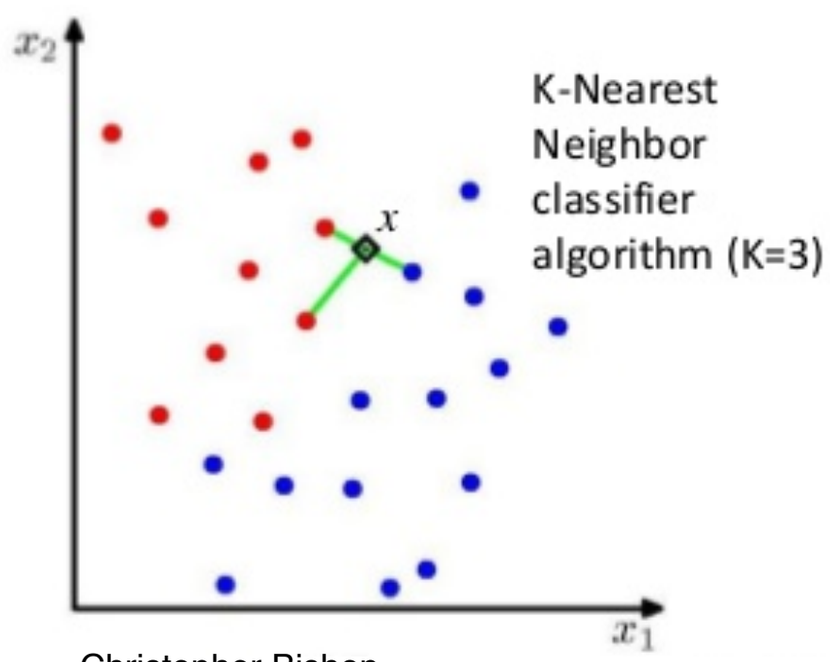


Christopher Bishop

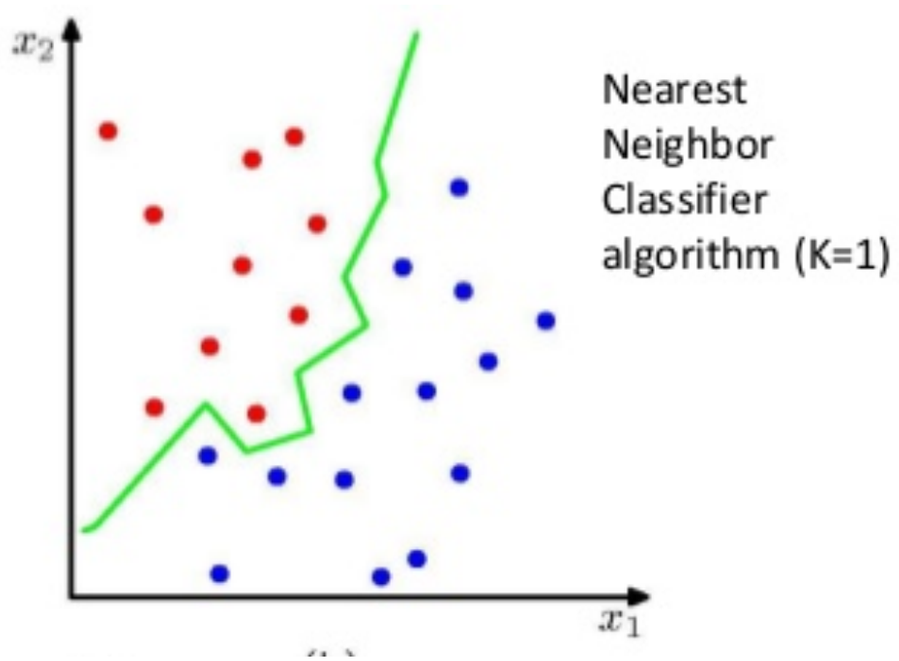
k-Nearest-neighbor classifier

- Psychologists: we learn from prototypes

→ Classify based on distance to nearest neighbors



Christopher Bishop

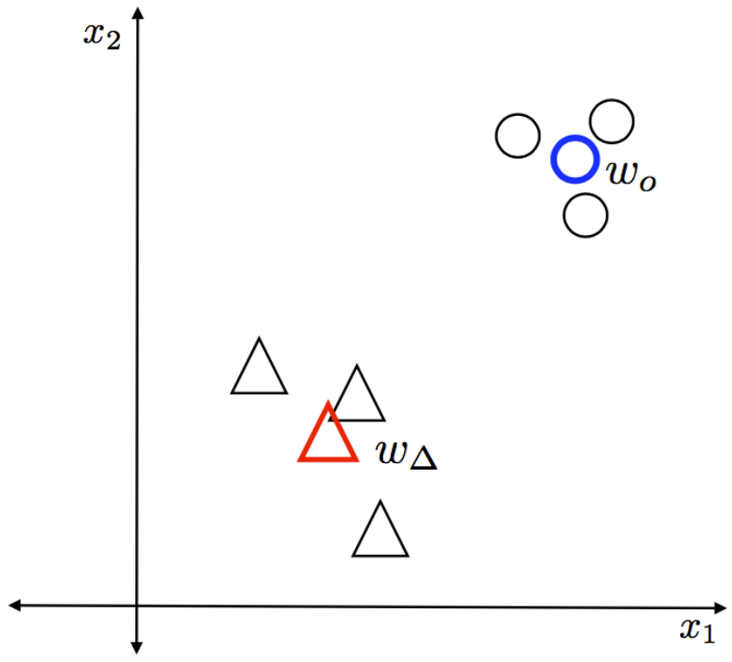


k-nearest-neighbor classifier

- Performs quite well in theory and practice
- Non-linear (e.g. piecewise-constant) decision boundary
- Computationally expensive

Nearest-centroid classifier

Classify based on distance to nearest class means



Class means $\boldsymbol{\mu}_\Delta$ and $\boldsymbol{\mu}_o$ as prototypes

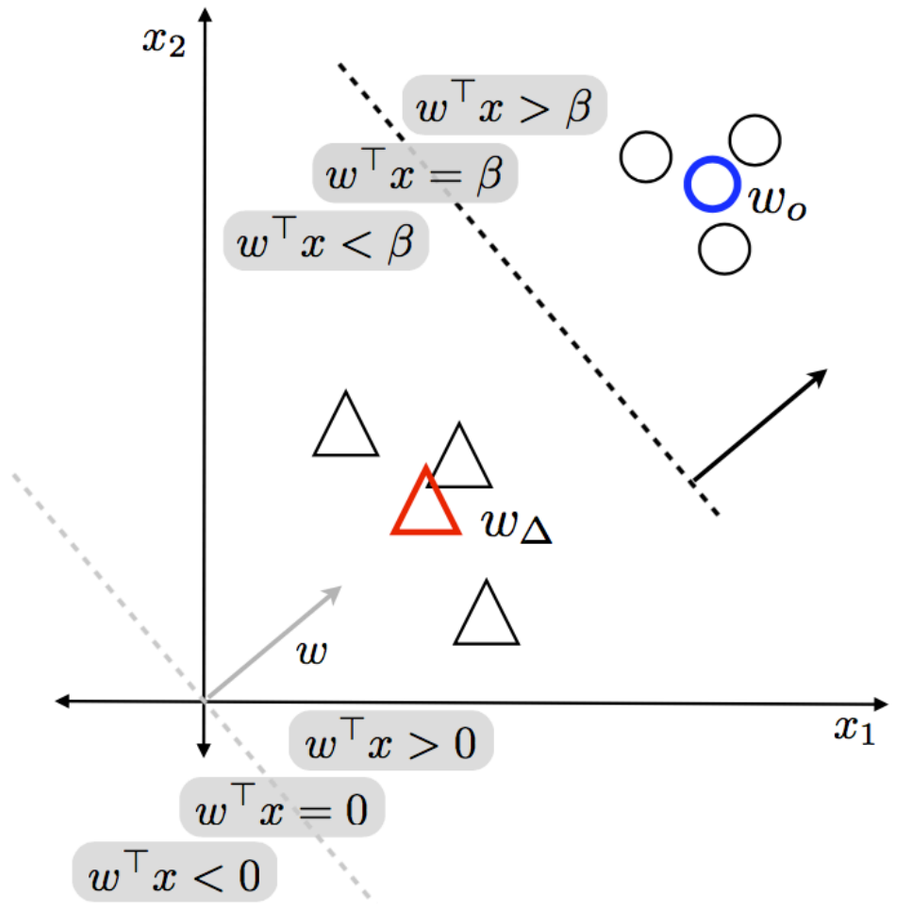
$$\boldsymbol{\mu}_\Delta = \frac{1}{N_\Delta} \sum_{i=1}^{N_\Delta} \mathbf{x}_{\Delta i}$$

$$\boldsymbol{\mu}_o = \frac{1}{N_o} \sum_{i}^{N_o} \mathbf{x}_{oi}$$

Distance from $\boldsymbol{\mu}_\Delta$ to new data $\mathbf{x}$

$$\|\boldsymbol{\mu}_\Delta - \mathbf{x}\| = \sqrt{\sum_{j=1}^{d} (\mu_{\Delta j} - x_j)^2}$$

Nearest-centroid classifier



Comparison of distance to class means is equivalent to linear classification

$$\|\mathbf{x} - \boldsymbol{\mu}_\Delta\| > \|\mathbf{x} - \boldsymbol{\mu}_o\|$$

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

where

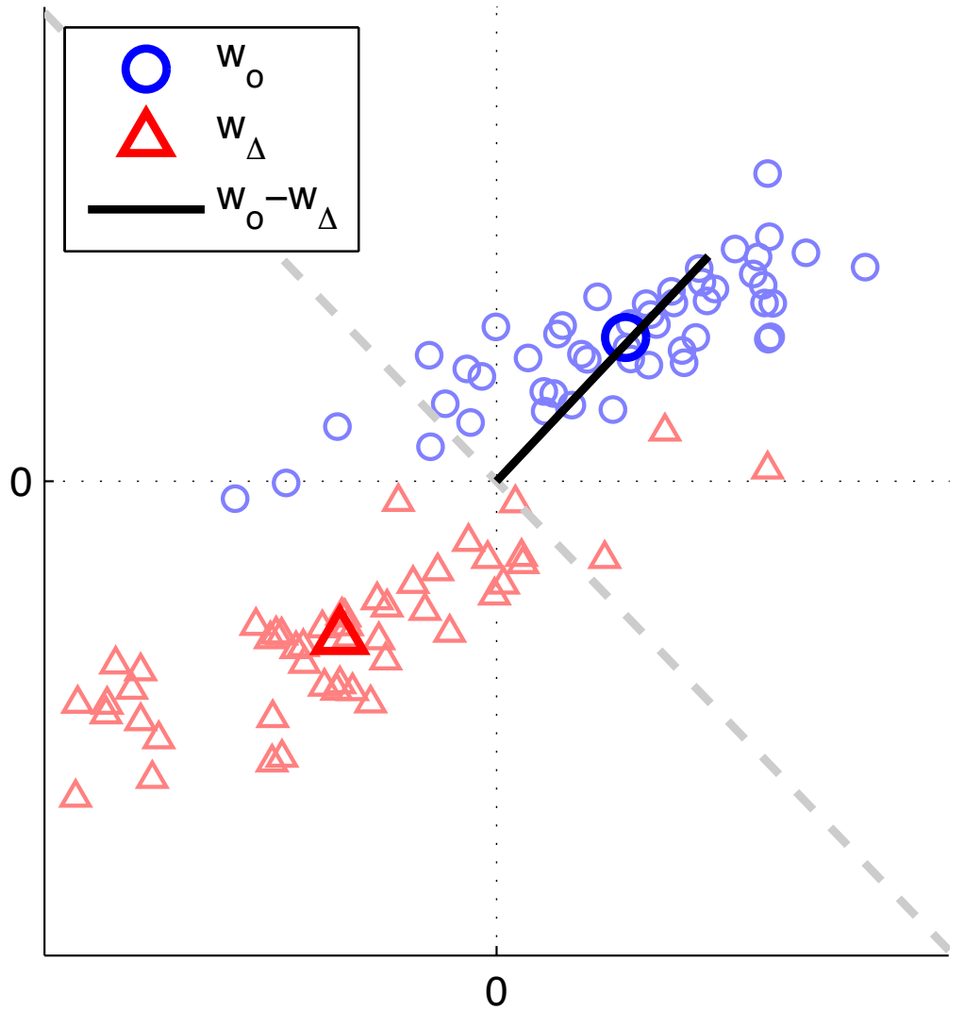
$$\mathbf{w} = \boldsymbol{\mu}_o - \boldsymbol{\mu}_\Delta$$

and

$$\beta = 1/2 \cdot (\boldsymbol{\mu}_o^\top \boldsymbol{\mu}_o - \boldsymbol{\mu}_\Delta^\top \boldsymbol{\mu}_\Delta)$$

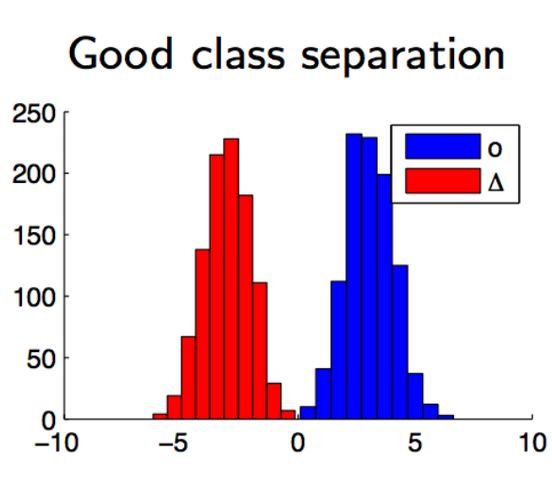
$$= 1/2 \cdot \mathbf{w}^\top (\boldsymbol{\mu}_o + \boldsymbol{\mu}_\Delta)$$

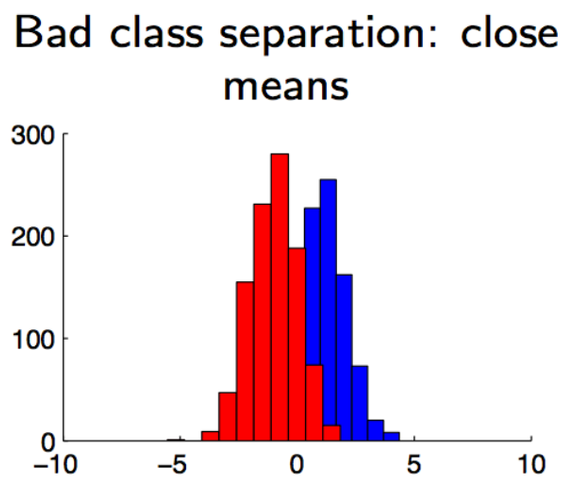
Problem: correlated features

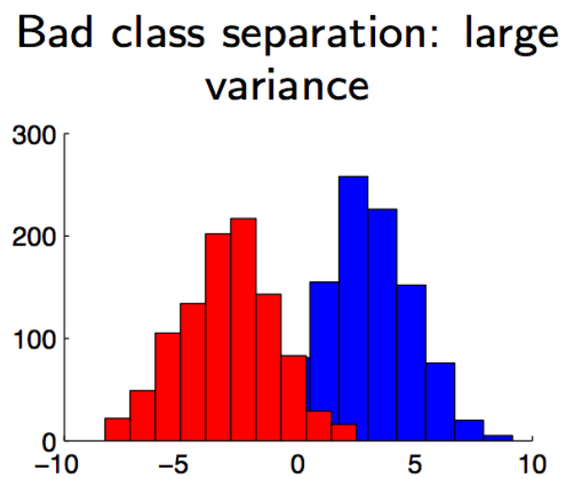


Class separability

- Consider one dimensional data and two classes



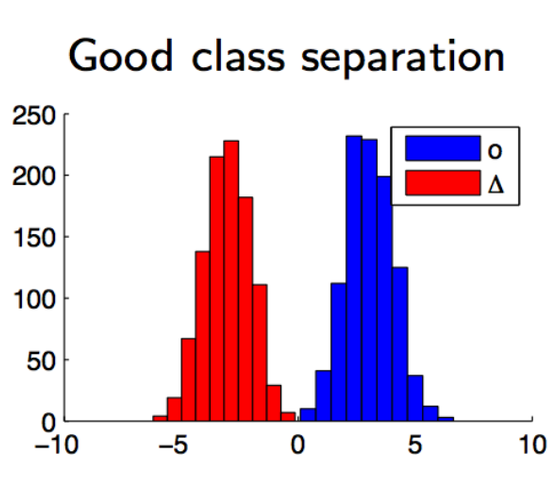


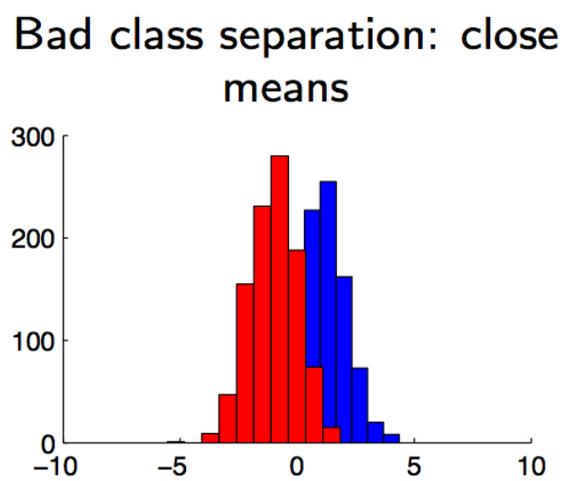


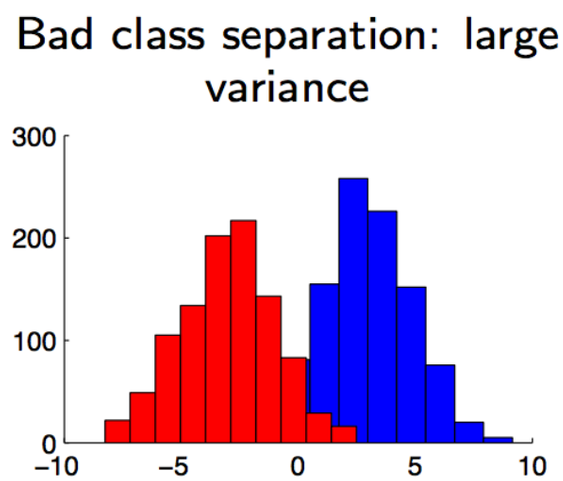
What is a good criterion for class separability?

Class separability

- Consider one dimensional data and two classes







What is a good criterion for class separability?

Fisher's criterion:

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

Square of Student-t statistic

Where $x_{o1}, \ldots, x_{oN_o} \in \mathbb{R}^1$, $\mu_o = \frac{1}{N_o} \sum_{i=1}^{N_o} x_{oi}$, and $\sigma_o^2 = \frac{1}{N_o} \sum_{i=1}^{N_o} (x_{oi} - \mu_o)^2$.

Ronald A. Fisher

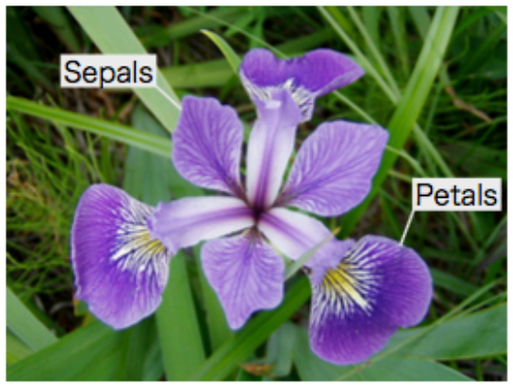
R.A. Fisher (1890 - 1962)

Founder of modern statistics

The *Iris* flower dataset

Iris Setosa

Iris Versicolor



Iris Virginica

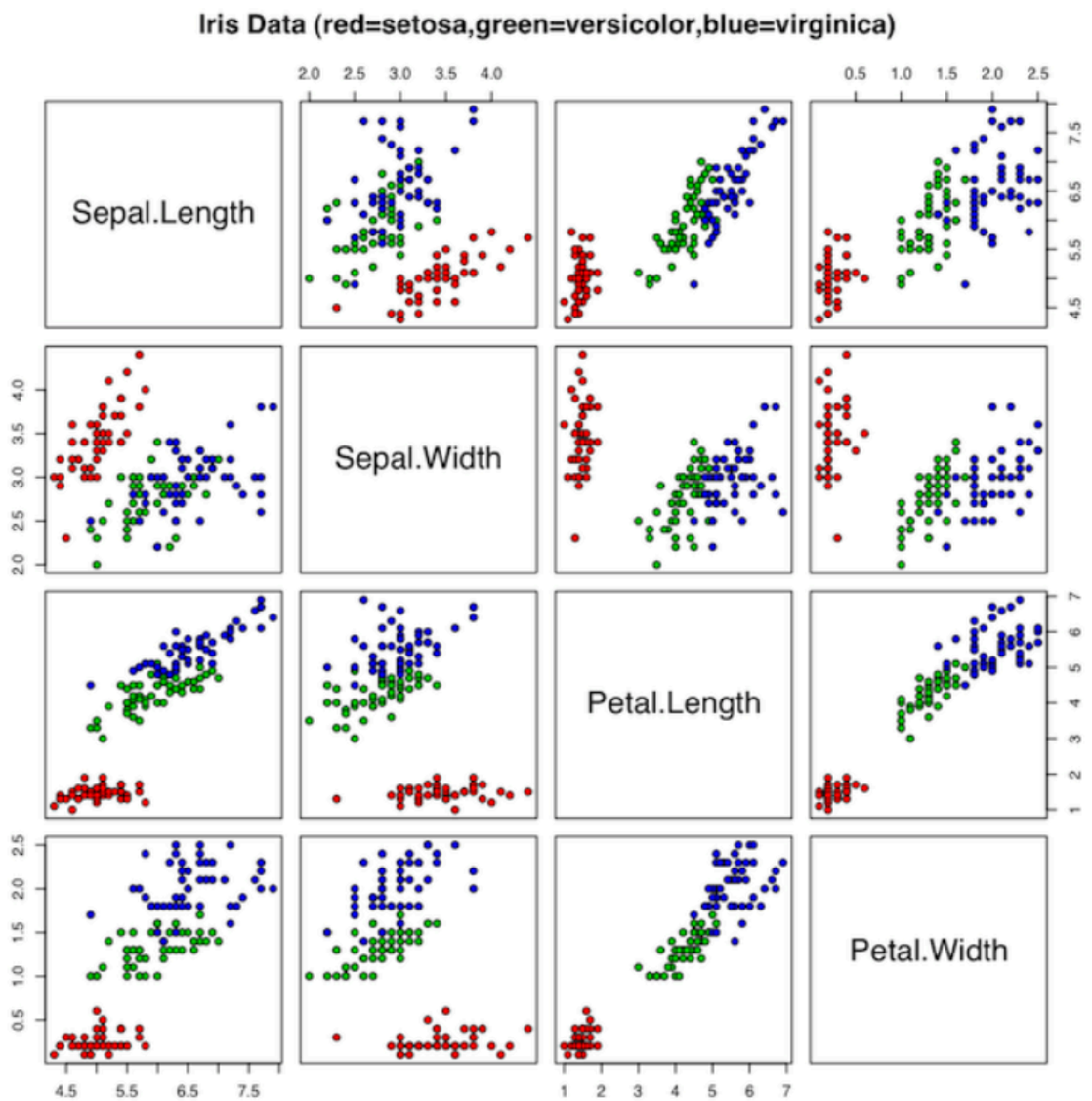
`http://en.wikipedia.org/wiki/Iris_flower_data_set`

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



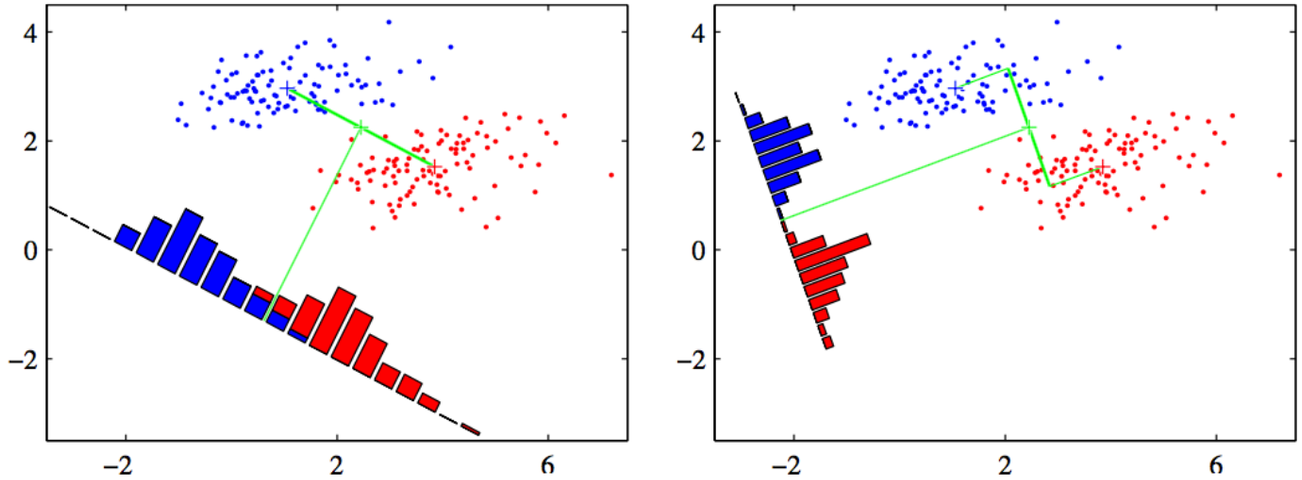
http://en.wikipedia.org/wiki/Iris_flower_data_set

Fisher's linear discriminant

Now consider d-dimensional data $\mathbf{x}_{o1}, \ldots, \mathbf{x}_{oN_o} \in \mathbb{R}^d$,

$\boldsymbol{\mu}_o = \frac{1}{N_o} \sum_{i=1}^{N_o} \mathbf{x}_{oi}$, and $\boldsymbol{\Sigma}_o = \frac{1}{N_o} \sum_{i=1}^{N_o} (\mathbf{x}_{oi} - \boldsymbol{\mu}_o)(\mathbf{x}_{oi} - \boldsymbol{\mu}_o)^\top$.

View classification in terms of dimensionality reduction (as in PCA)



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

Fisher's linear discriminant

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

Maximizes mean class difference

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

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

Fisher's linear discriminant

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

Maximizes mean class difference

$$(\mathbf{w}^\top \boldsymbol{\mu}_o - \mathbf{w}^\top \boldsymbol{\mu}_\Delta)^2 = \left(\mathbf{w}^\top (\boldsymbol{\mu}_o - \boldsymbol{\mu}_\Delta)\right)^2$$
$$= \mathbf{w}^\top \underbrace{(\boldsymbol{\mu}_o - \boldsymbol{\mu}_\Delta)(\boldsymbol{\mu}_o - \boldsymbol{\mu}_\Delta)^\top}_{S_B - \text{"between-class scatter matrix"}} \mathbf{w}$$

Minimizes variance in each class

$$\frac{1}{N_o}\sum_{i=1}^{N_o}\left(\mathbf{w}^\top(\mathbf{x}_{oi} - \boldsymbol{\mu}_o)\right)^2 + \frac{1}{N_\Delta}\sum_{j=1}^{N_\Delta}\left(\mathbf{w}^\top(\mathbf{x}_{\Delta j} - \boldsymbol{\mu}_\Delta)\right)^2$$
$$= \mathbf{w}^\top \underbrace{\left(\frac{1}{N_o}\sum_{i=1}^{N_o}(\mathbf{x}_{oi} - \boldsymbol{\mu}_o)(\mathbf{x}_{oi} - \boldsymbol{\mu}_o)^\top + \frac{1}{N_\Delta}\sum_{j=1}^{N_\Delta}(\mathbf{x}_{\Delta j} - \boldsymbol{\mu}_\Delta)(\mathbf{x}_{\Delta j} - \boldsymbol{\mu}_\Delta)^\top\right)}_{S_W - \text{"within-class scatter matrix"}} \mathbf{w}$$

Fisher's linear discriminant

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

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

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

$\rightarrow$ FLD solution: $\mathbf{w}$ that optimizes the Fisher criterion

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

Generalized Rayleigh Quotient

Fisher's linear discriminant

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\text{argmax}} \frac{\mathbf{w}^\top S_B \mathbf{w}}{\mathbf{w}^\top S_W \mathbf{w}}$$

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

Fisher's linear discriminant

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\text{argmax}} \frac{\mathbf{w}^\top S_B \mathbf{w}}{\mathbf{w}^\top S_W \mathbf{w}}$$

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

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

Lagrange Multiplier = Rayleigh Quotient at Optimum

Generalized Eigenvalue problem!

Fisher's linear discriminant

- For invertible S_W, problem reduces to regular eigenvalue problem:

$$\lambda \mathbf{w} = S_W^{-1} S_B \mathbf{w} \qquad (S_B \text{ is always non-invertible})$$

Fisher's linear discriminant

- For invertible S_W, problem reduces to regular eigenvalue problem:

$$\lambda \mathbf{w} = S_W^{-1} S_B \mathbf{w} \qquad (S_B \text{ is always non-invertible})$$

- A direct solution uses that $S_B = (\boldsymbol{\mu}_o - \boldsymbol{\mu}_\Delta)(\boldsymbol{\mu}_o - \boldsymbol{\mu}_\Delta)^\top$ has rank 1

$$\lambda S_W \mathbf{w} = (\boldsymbol{\mu}_o - \boldsymbol{\mu}_\Delta) \underbrace{(\boldsymbol{\mu}_o - \boldsymbol{\mu}_\Delta)^\top \mathbf{w}}_{\text{scalar}}$$

$$S_W \mathbf{w} = (\boldsymbol{\mu}_o - \boldsymbol{\mu}_\Delta) c$$

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

Fisher's linear discriminant

- For invertible S_W, problem reduces to regular eigenvalue problem:

$$\lambda \mathbf{w} = S_W^{-1} S_B \mathbf{w} \qquad (S_B \text{ is always non-invertible})$$

- A direct solution uses that $S_B = (\boldsymbol{\mu}_o - \boldsymbol{\mu}_\Delta)(\boldsymbol{\mu}_o - \boldsymbol{\mu}_\Delta)^\top$ has rank 1

$$\lambda S_W \mathbf{w} = (\boldsymbol{\mu}_o - \boldsymbol{\mu}_\Delta) \underbrace{(\boldsymbol{\mu}_o - \boldsymbol{\mu}_\Delta)^\top \mathbf{w}}_{\text{scalar}}$$

$$S_W \mathbf{w} = (\boldsymbol{\mu}_o - \boldsymbol{\mu}_\Delta) c$$

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

→ $\mathbf{w} = S_B^{-1}(\boldsymbol{\mu}_o - \boldsymbol{\mu}_\Delta)$ is Bayes optimal if $\mathbf{x}_o \sim \mathcal{N}(\boldsymbol{\mu}_o, 1/2 S_W)$, $\mathbf{x}_\Delta \sim \mathcal{N}(\boldsymbol{\mu}_\Delta, 1/2 S_W)$

- Use LDA bias $b = \frac{1}{2} \mathbf{w}^\top (\boldsymbol{\mu}_o + \boldsymbol{\mu}_\Delta) + \log \frac{p(\Delta)}{p(o)}$

Fisher's linear discriminant

- Similarity to PCA: for $S_B = S = \frac{1}{N} X X^\top$ and $S_W = I$, PCA is obtained

Fisher's linear discriminant

- Similarity to PCA: for $S_B = S = \frac{1}{N} X X^\top$ and $S_W = I$, PCA is obtained

→ Possible to derive FLD solution using Lagrange multipliers

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \mathbf{w}^\top \boldsymbol{S}_W \mathbf{w} \;\text{ s.t. }\; \mathbf{w}^\top \boldsymbol{S}_B \mathbf{w} = \text{const.}$$

$$= \arg\min_{\mathbf{w}} \mathbf{w}^\top \boldsymbol{S}_W \mathbf{w} \;\text{ s.t. }\; \left(\mathbf{w}^\top (\mu_o - \mu_\Delta)\right)^2 = \text{const.}$$

- λ = generalized Lagrange multiplier = ratio of between and within-class variance = Fisher criterion at optimum

Fisher's linear discriminant

- Similarity to PCA: for $S_B = S = \frac{1}{N} X X^\top$ and $S_W = I$, PCA is obtained

→ Possible to derive FLD solution using Lagrange multipliers

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \mathbf{w}^\top \boldsymbol{S}_W \mathbf{w} \;\text{ s.t. }\; \mathbf{w}^\top \boldsymbol{S}_B \mathbf{w} = \text{const.}$$

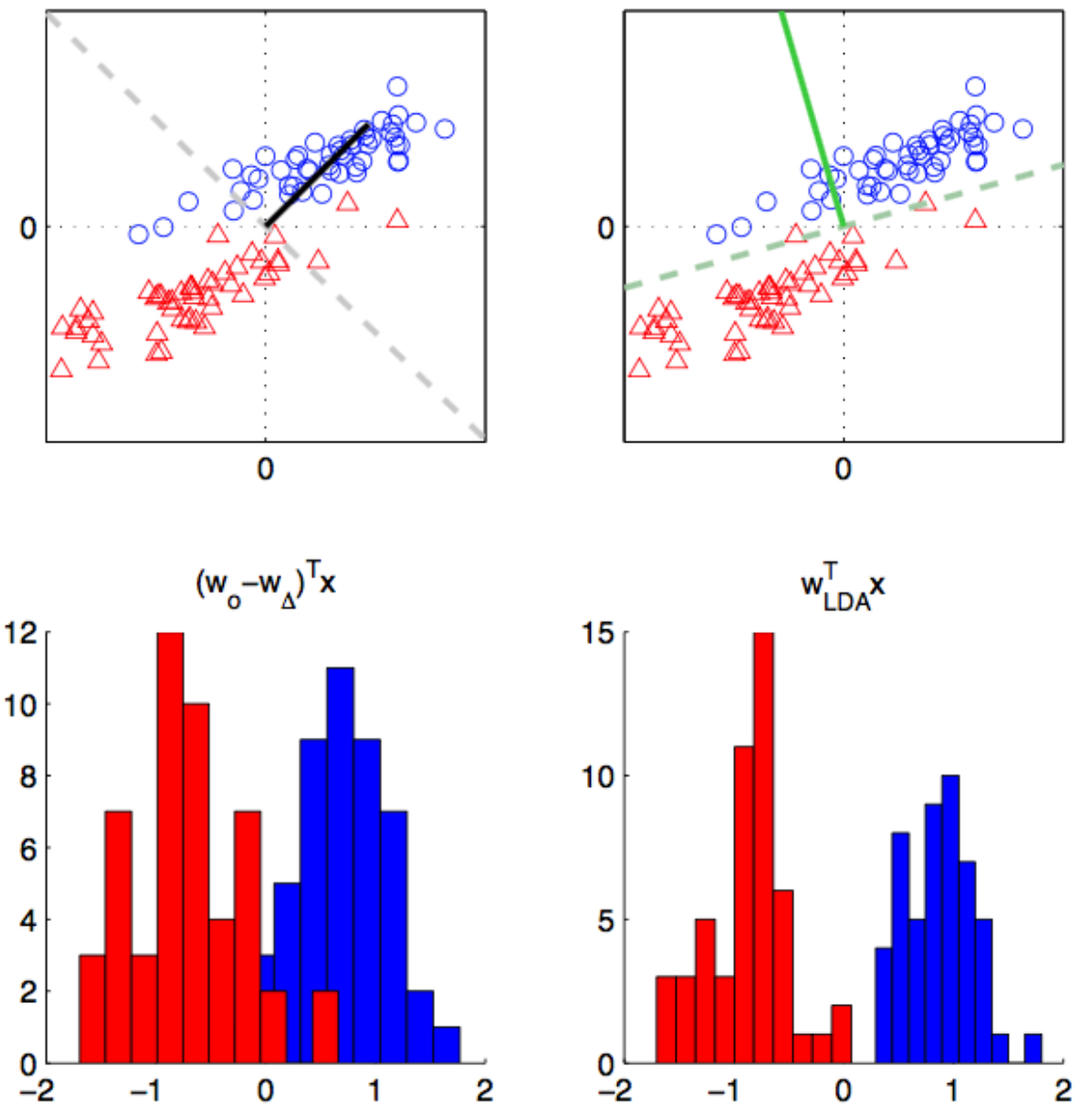
$$= \arg\min_{\mathbf{w}} \mathbf{w}^\top \boldsymbol{S}_W \mathbf{w} \;\text{ s.t. }\; \left(\mathbf{w}^\top (\mu_o - \mu_\Delta)\right)^2 = \text{const.}$$

- λ = generalized Lagrange multiplier = ratio of between and within-class variance = Fisher criterion at optimum

→ Possible to derive PCA solution as maximum of Rayleigh quotient

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \frac{\mathbf{w}^\top \boldsymbol{S} \mathbf{w}}{\mathbf{w}^\top \mathbf{w}}$$

Fisher's linear discriminant vs. nearest-centroid classifier



Another view on FLD

An equivalent formulation of LDA is given by decomposing the total covariance of the data as $S = S_W + \frac{N_\Delta N_o}{(N_\Delta + N_o)} S_B$. Then:

$$
\begin{aligned}
S_W \mathbf{w} &\propto S_B \mathbf{w} \\
(S - \frac{N_\Delta N_o}{N_\Delta + N_o} S_B)\mathbf{w} &\propto S_B \mathbf{w} \\
S\mathbf{w} &\propto S_B \mathbf{w} \\
\mathbf{w} &\propto S^{-1}(\boldsymbol{\mu}_o - \boldsymbol{\mu}_\Delta) \,.
\end{aligned}
$$

Another view on FLD

FLD decorrelates the data followed by nearest centroid classification:

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

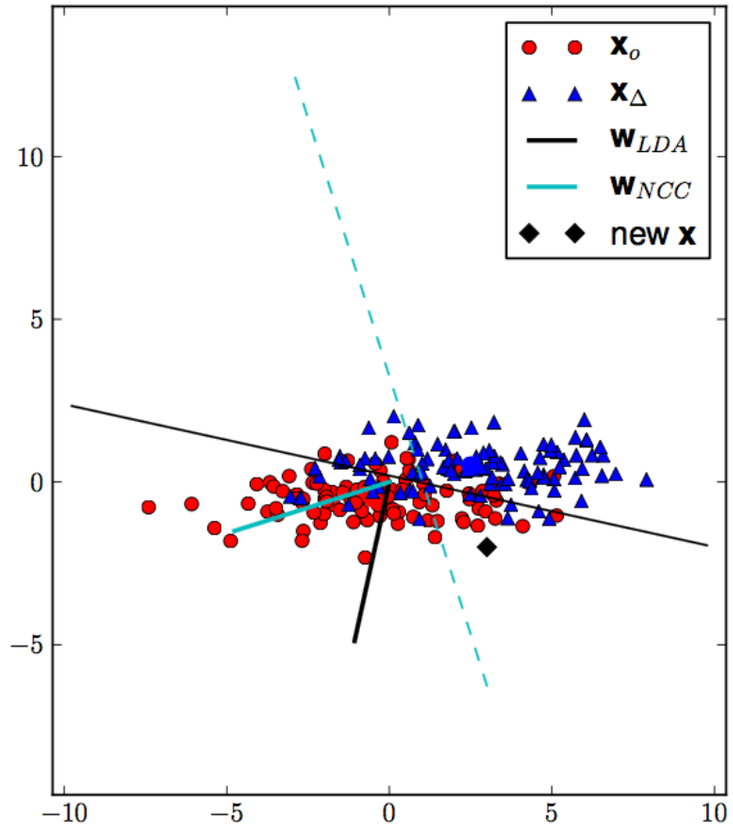
$$\mathbf{w}^T\mathbf{x} = (\boldsymbol{\mu}_o - \boldsymbol{\mu}_\Delta)^T S^{-1}\mathbf{x} = \underbrace{(\boldsymbol{\mu}_o - \boldsymbol{\mu}_\Delta)^T U\Lambda^{-1/2}}_{\substack{\text{mean class difference} \\ \text{of decorrelated data}}} \underbrace{\Lambda^{-1/2}U^T\mathbf{x}}_{\text{decorrelated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

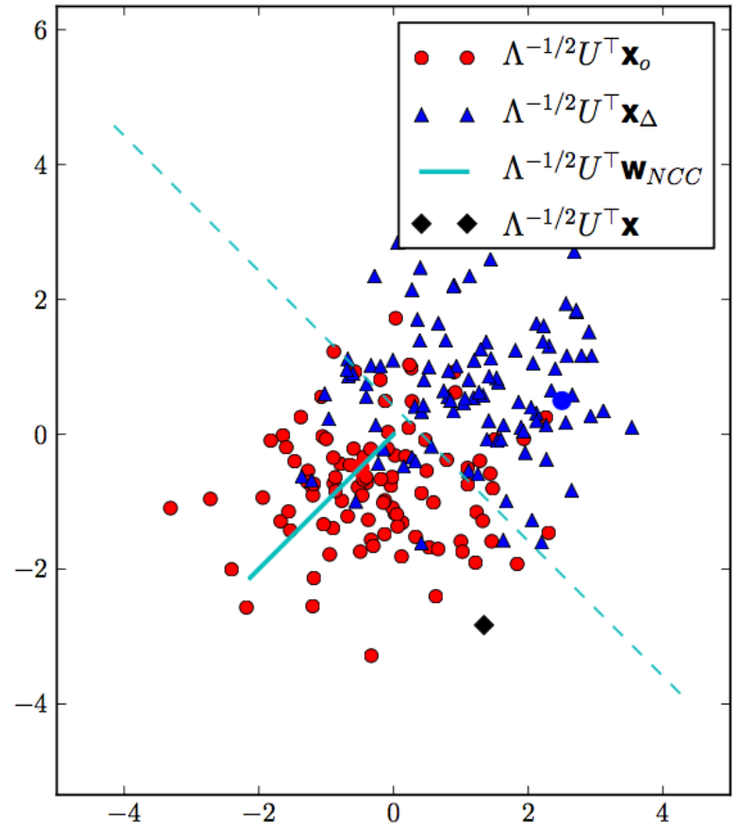
where $S = U\Lambda U^T$ is the eigenvalue decomposition (PCA) of S.

Another view on FLD

FLD decorrelates the data followed by nearest centroid classification:

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





Multi-class FLD

$$S_W = \frac{1}{C}\sum_{i=1}^{C} S_i$$

$$S_B = \frac{1}{C}\sum_{i=1}^{C} (\mu_i - \mu)(\mu_i - \mu)^\top$$

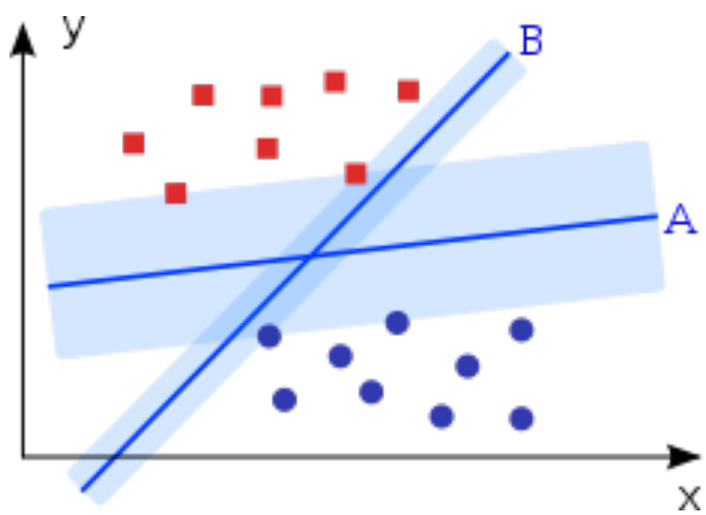
$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \frac{\mathbf{w}^\top S_B \mathbf{w}}{\mathbf{w}^\top S \mathbf{w}}$$

Leads to *C-1* different components $\hat{w}_1, \ldots, \hat{w}_{C-1}$

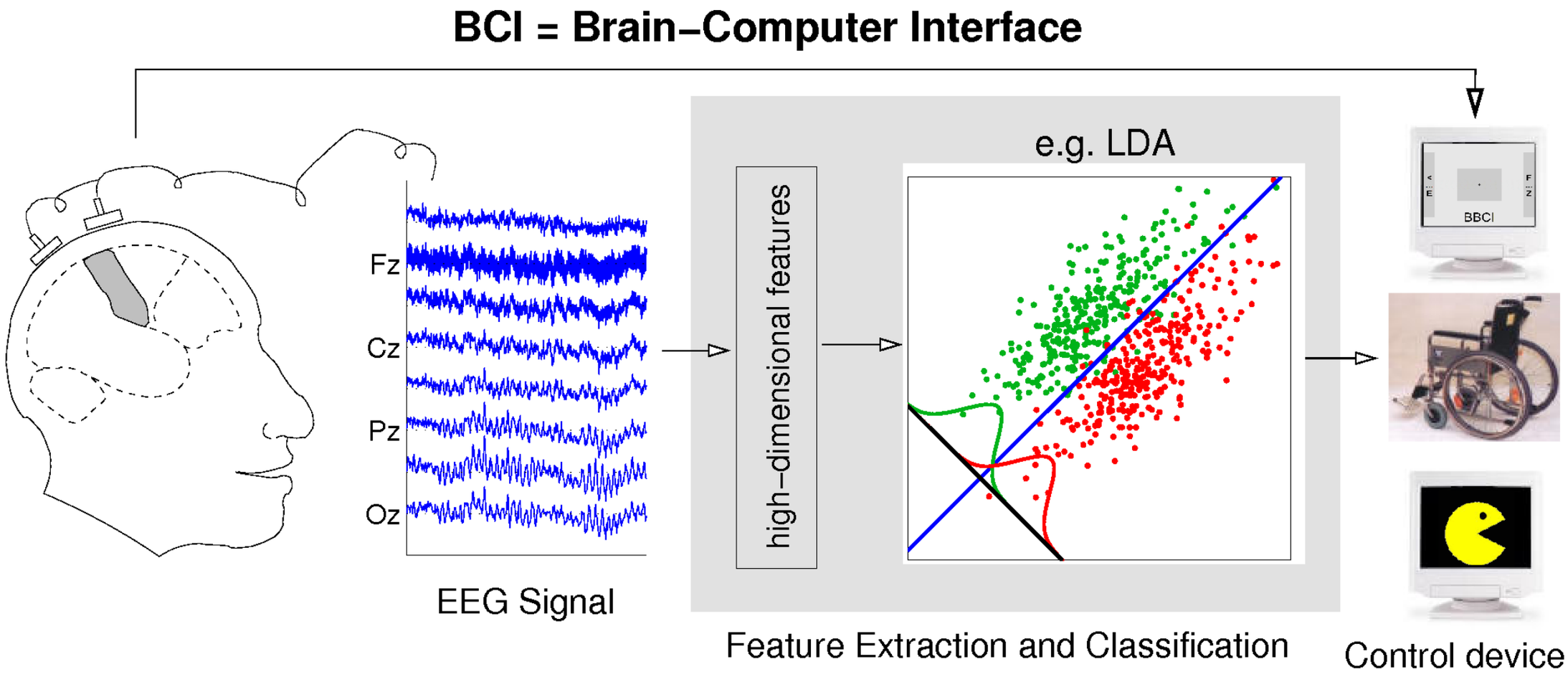
Other linear classifiers

- LDA/FDA are based on covariance matrices
  - Inherent Gaussianity assumption
  - Solution based on inverse of $d \times d$ covariance matrix

- Desirable to avoid explicit covariance estimation/inversion
  - Support vector machines
  - Logistic regression



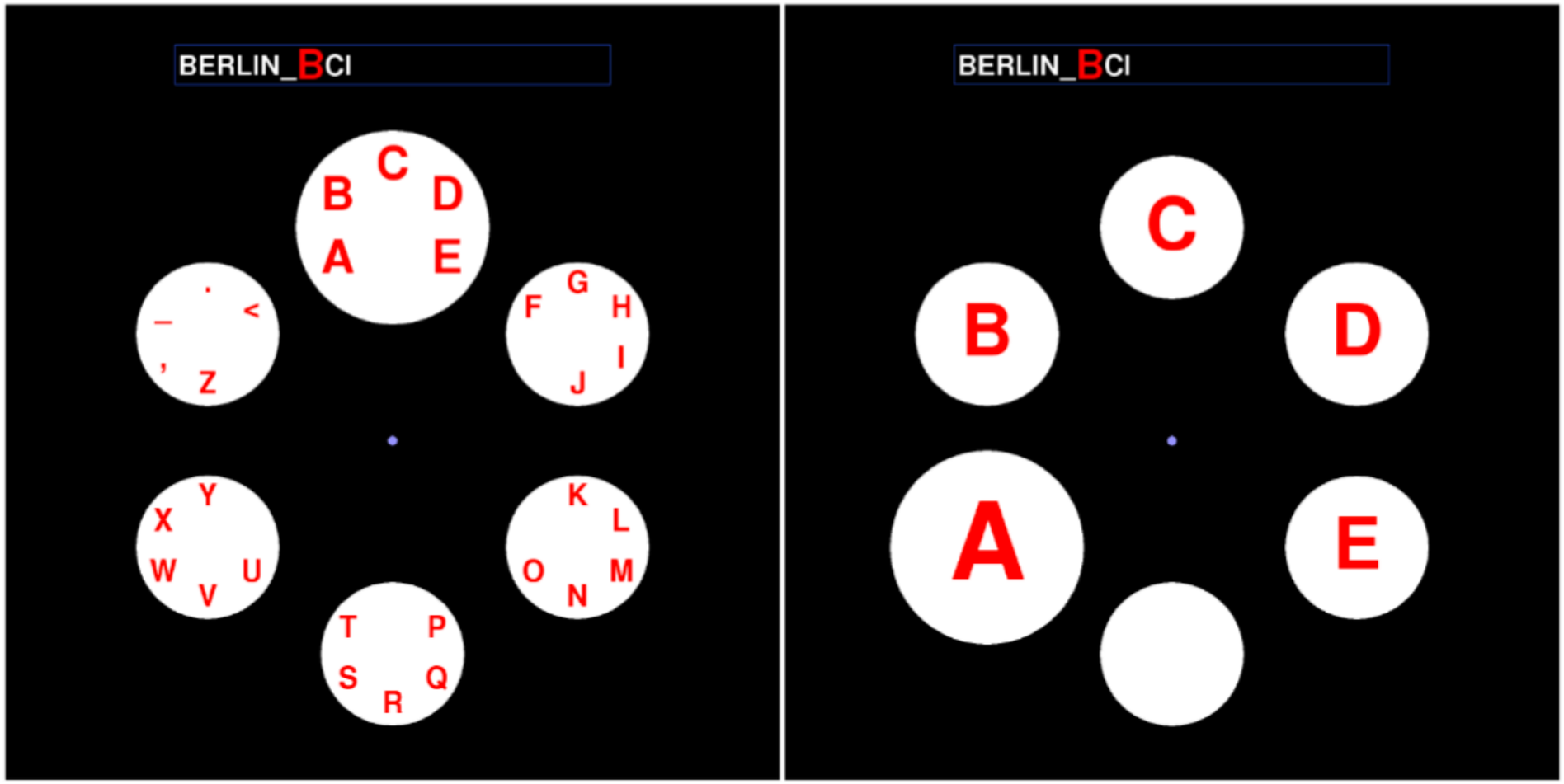
Application: Brain-computer interfaces



BCI Speller

Hex-o-spell: Writing with thoughts

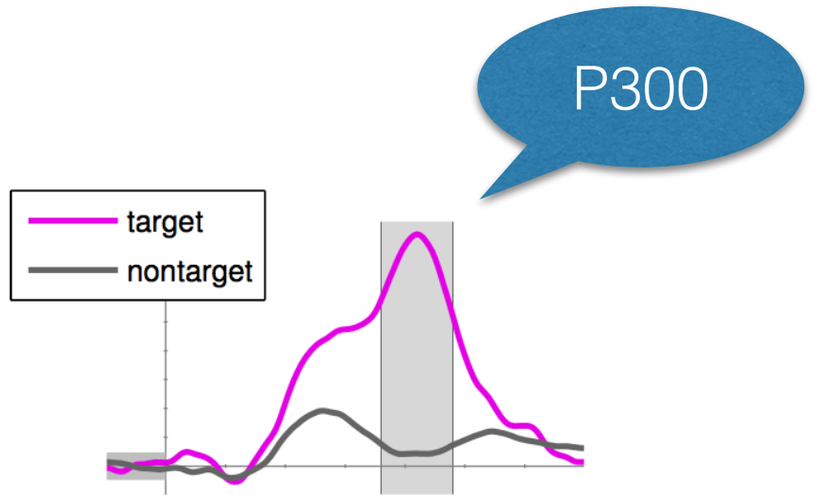
http://www.bbci.de/

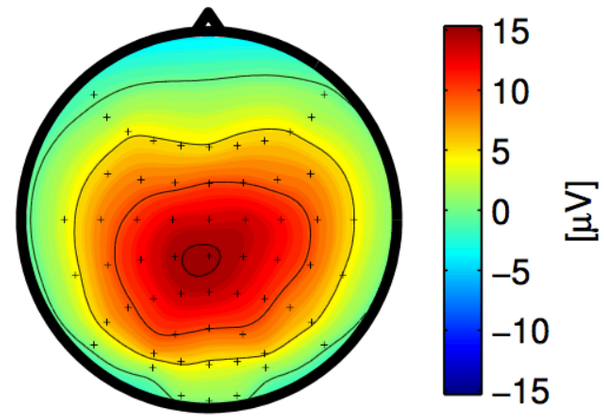


Demo: http://iopscience.iop.org/1741-2552/8/6/066003/media

BCI Speller

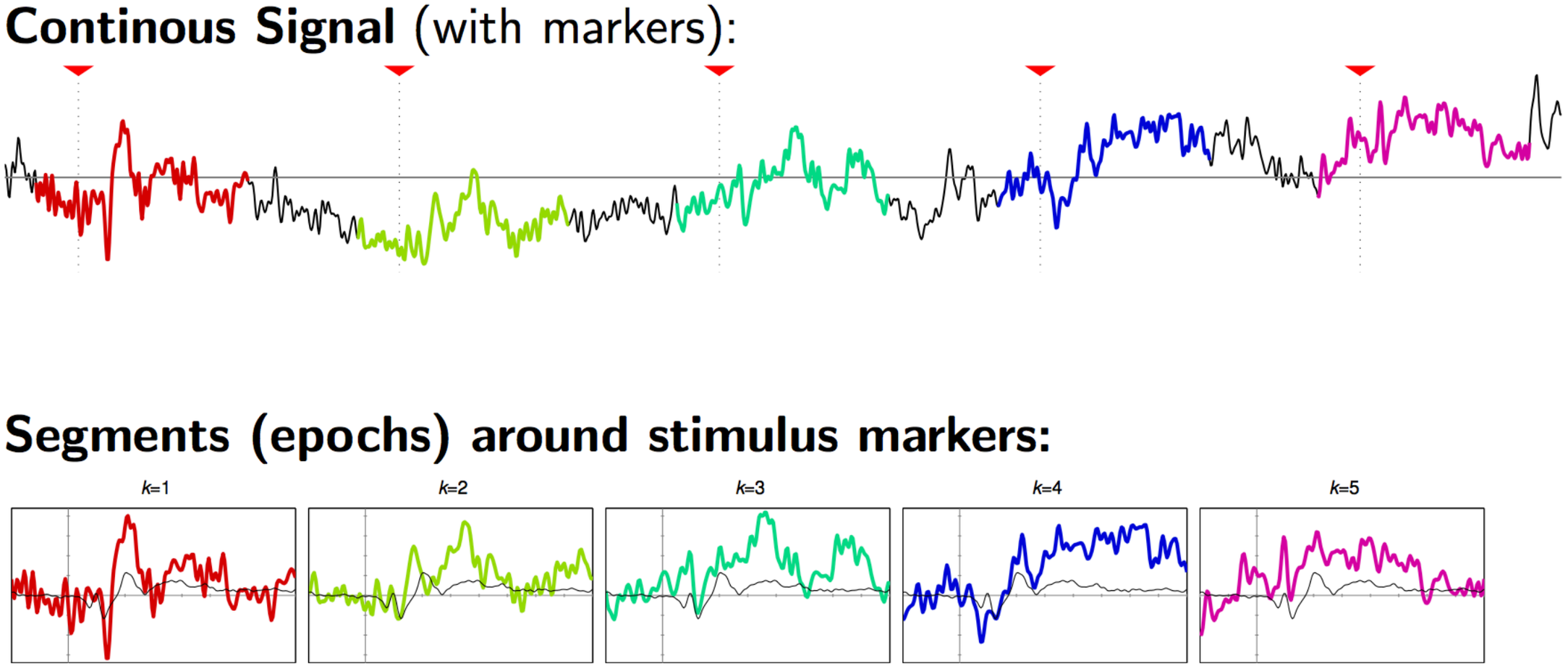
- User concentrates on a symbol (or symbol group)
- Symbols are intensified randomly
- Target symbols elicit specific ERPs
- BCI detects target ERPs (averaged over few repetitions)



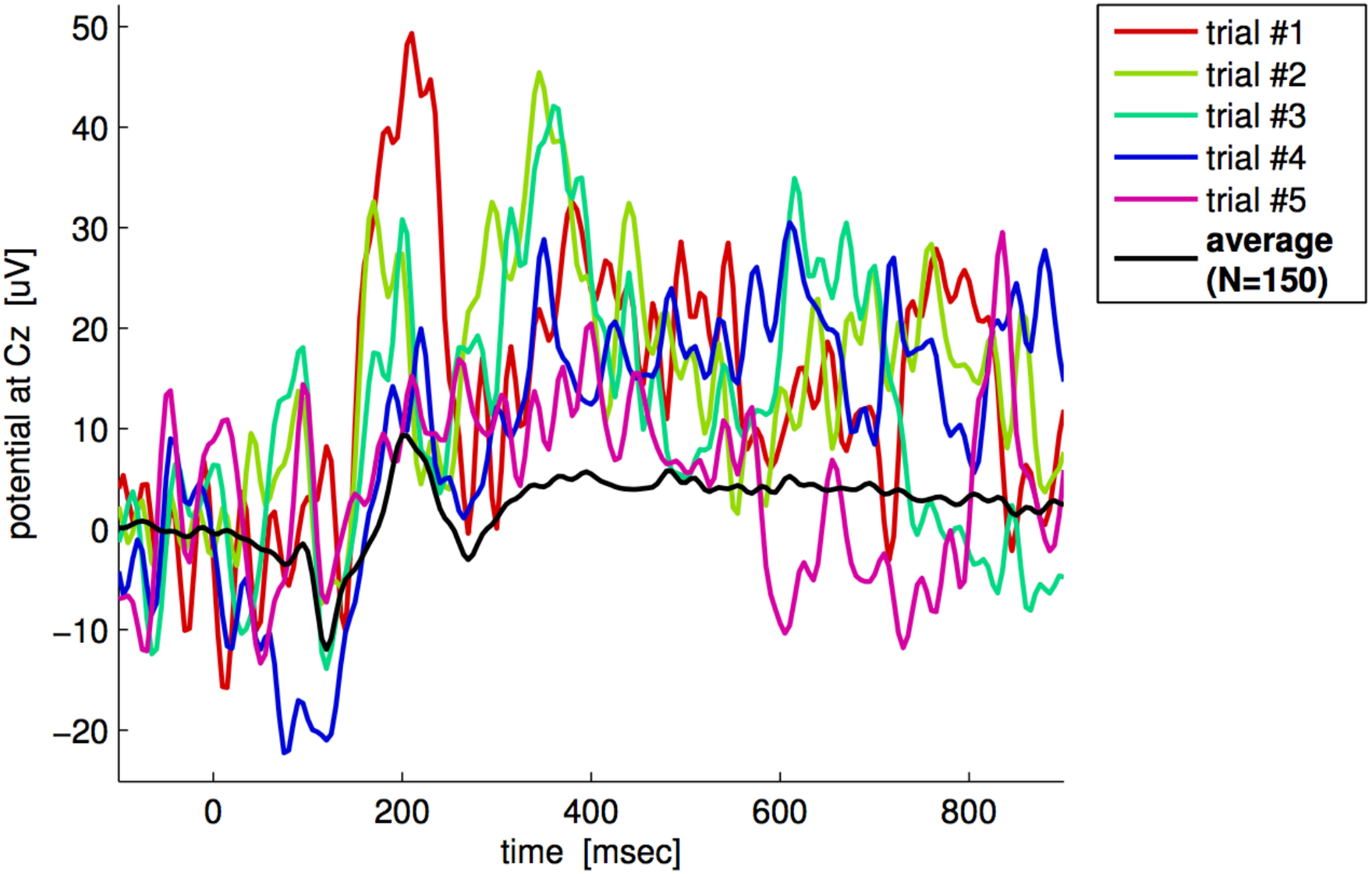


ERP = event-related (brain electrical) potential

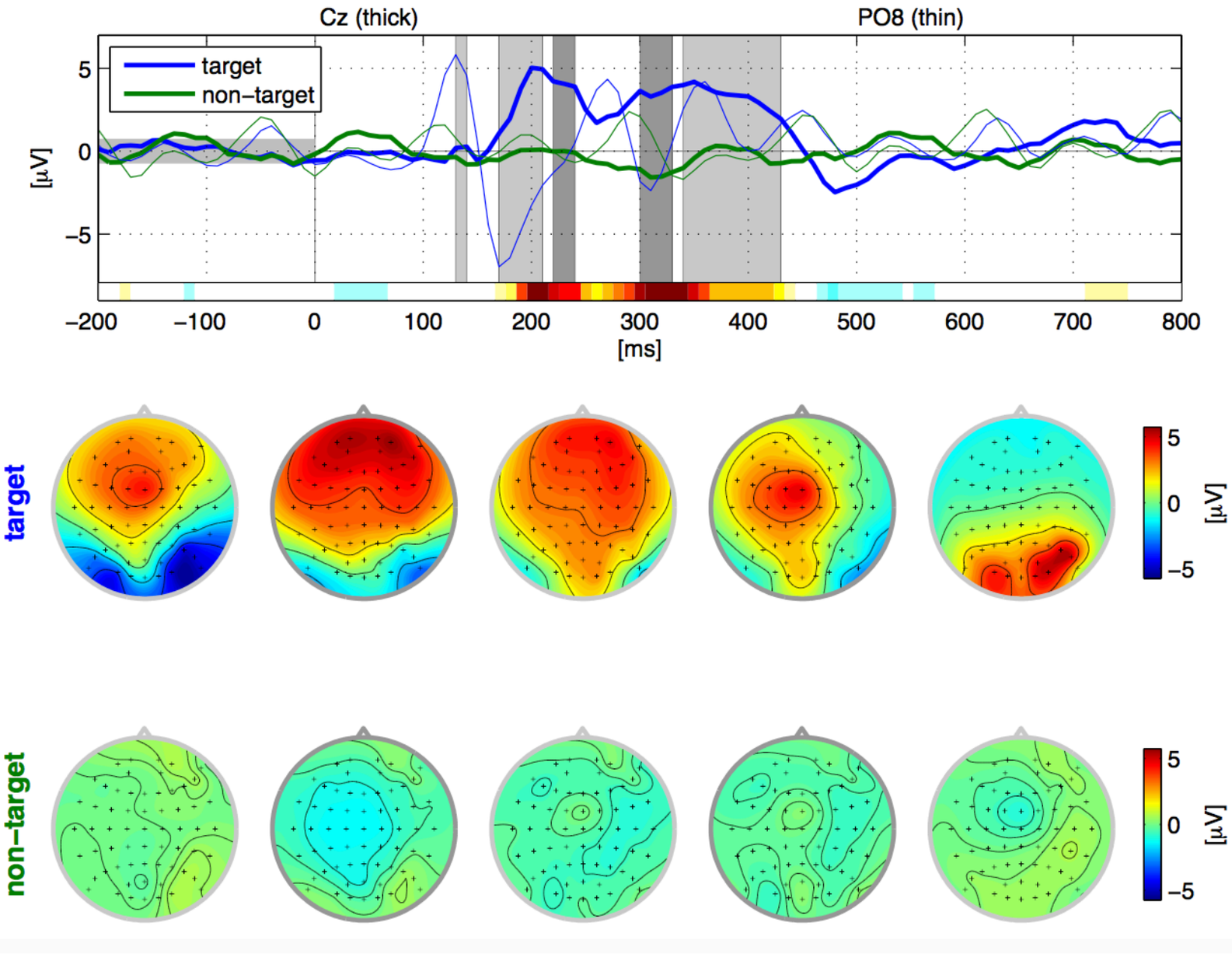
Event-related potentials



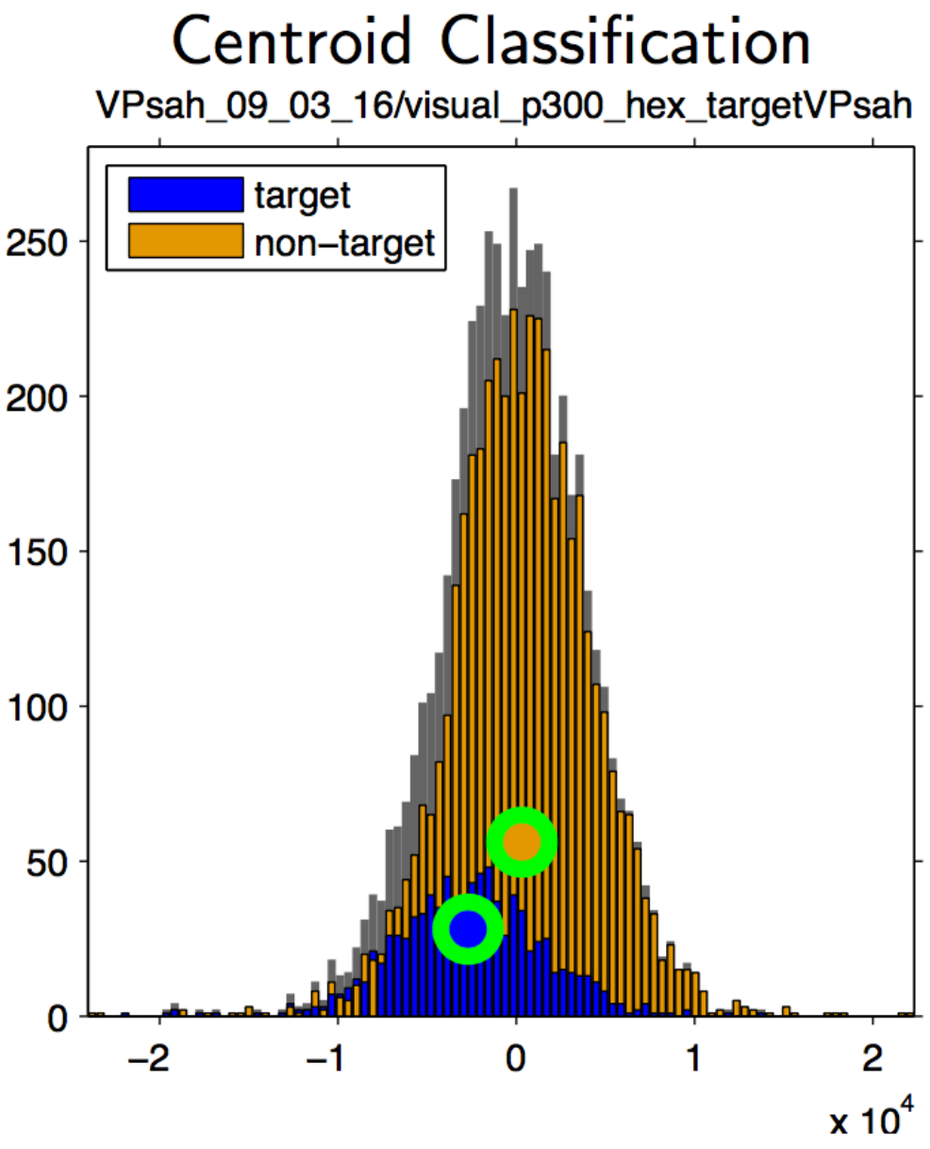
Event-related potentials

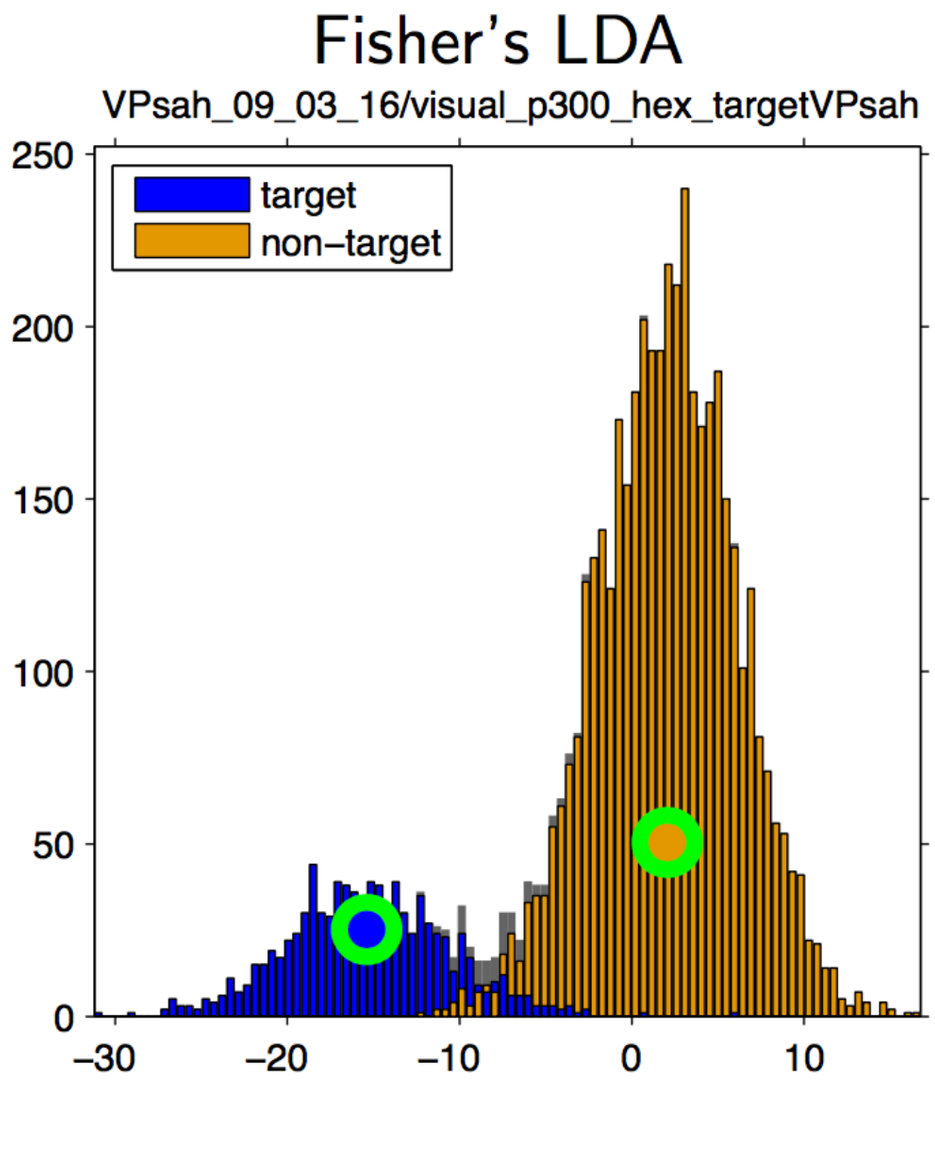


Target vs. non-target stimuli

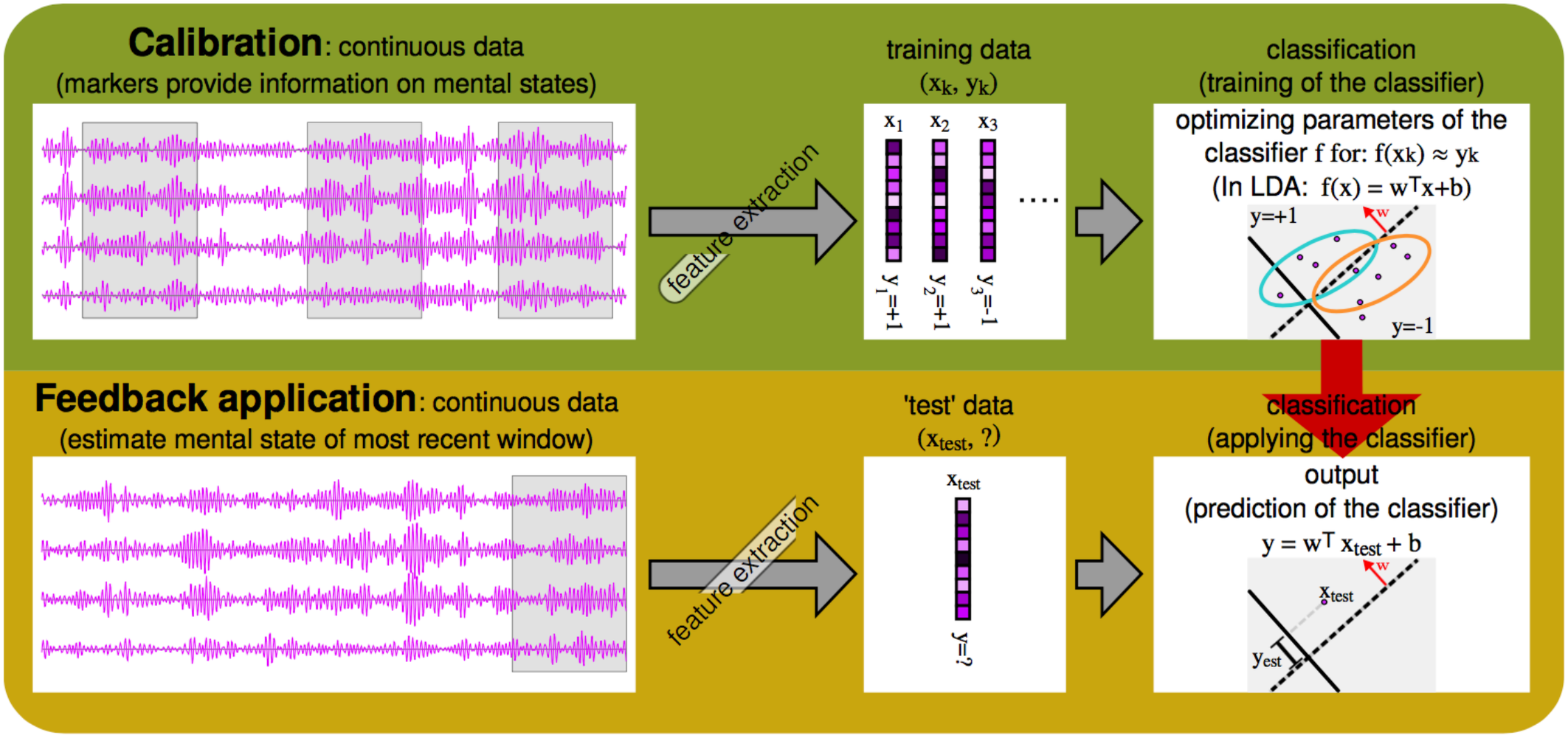


Separability





Calibration and application



https://www.youtube.com/watch?v=kBjlftqoctM

Component analysis

- More general view on discriminative models:

  find data representation that captures “interesting” properties

$$\hat{W} = \arg\max_{W} f(W^\top X)$$

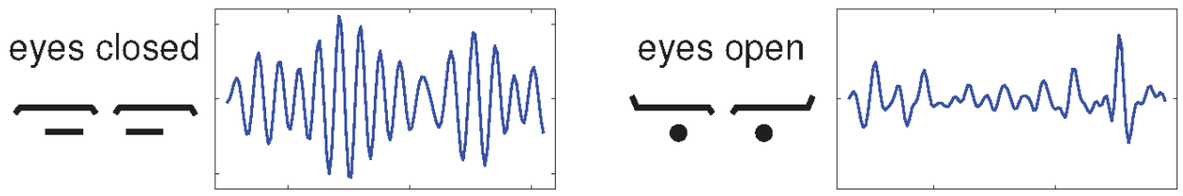
- Examples:
  - Independent Component Analysis (ICA)
  - Canonical Correlation Analysis (CCA)
  - Common Spatial Patterns (CSP)
  - Stationary Subspace Analysis (SSA)
  - Spatio-spectral Decomposition (SSD)
  - …

Looking for oscillations

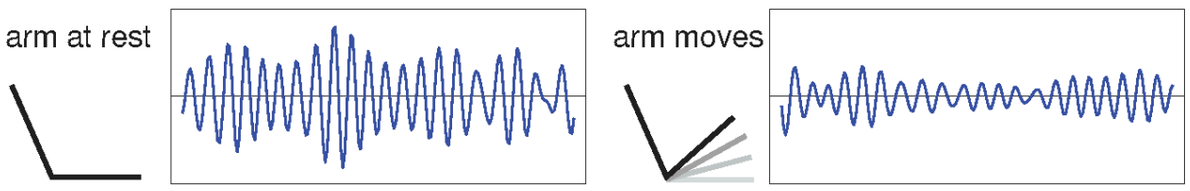
Not all EEG phenomena are phase-locked to certain events. There are also rhythms, the amplitude of which modulates depending on the mental state.

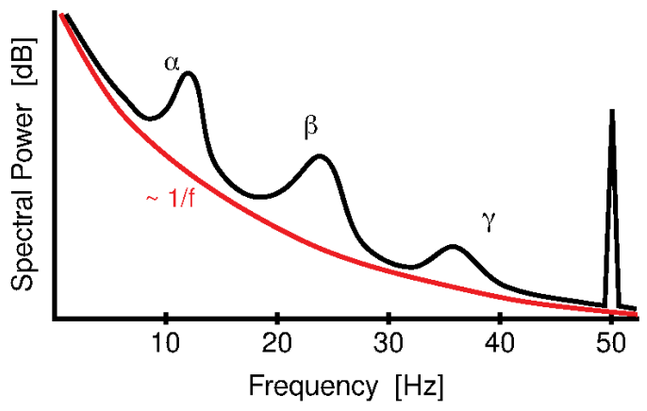
Most rhythms are idle rhythms, i.e., are attenuated during activation.

- α-rhythm (around 10 Hz) in visual cortex:



- μ-rhythm (around 10 Hz) in motor and sensory cortex:





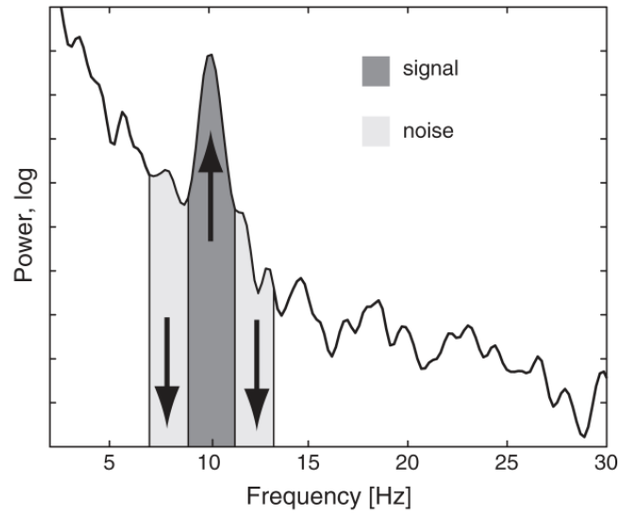
Figures by Benjamin Blankertz

Spatio-spectral decomposition

Signal of interest is narrow-band oscillation.

$$\mathbf{w} = \arg\max_{\mathbf{w}'} \mathrm{SNR}(\mathbf{w}')$$

$$= \arg\max_{\mathbf{w}'} \frac{\mathbf{w}'^{\top} \boldsymbol{\Sigma}_{\mathrm{signal}} \mathbf{w}'}{\mathbf{w}'^{\top} \left(\boldsymbol{\Sigma}_{\mathrm{noise}}\right) \mathbf{w}'}$$



$\boldsymbol{\Sigma}_{\mathrm{signal}}$ and $\boldsymbol{\Sigma}_{\mathrm{noise}}$ are the covariances of the data filtered in the central and flanking frequency bands.

$\mathbf{w}$ is obtained as the solution to the generalized eigenvalue equation

$\boldsymbol{\Sigma}_{\mathrm{signal}}\mathbf{w} = \lambda\boldsymbol{\Sigma}_{\mathrm{noise}}\mathbf{w}$ (in Matlab: $\mathbf{W} = \mathrm{eig}(\boldsymbol{\Sigma}_{\mathrm{signal}}, \boldsymbol{\Sigma}_{\mathrm{noise}});$) . [Nikulin et al., 2011]

Interpretation vs. SNR optimization

Two major goals

1. **Reconstruction:** Accurately estimate interesting (e.g. discriminative) components.

   Purpose: Predict class membership well.

   Example: Decode somebody's cognitive state from brain signals.

2. **Interpretation:** Identify data features that are related to a component.

   Purpose: Gain insight into problem.

   Example: Find out *where* in the brain a cognitive state is represented.

Possible to have both at the same time?

Linear backward model

Idea: combine channels to approximate class label.

$\mathbf{W}^\top \mathbf{x}(t) = \mathbf{s}(t)$, where

the columns of $\mathbf{W} \in \mathbb{R}^{M \times K}$ are called extraction filters.

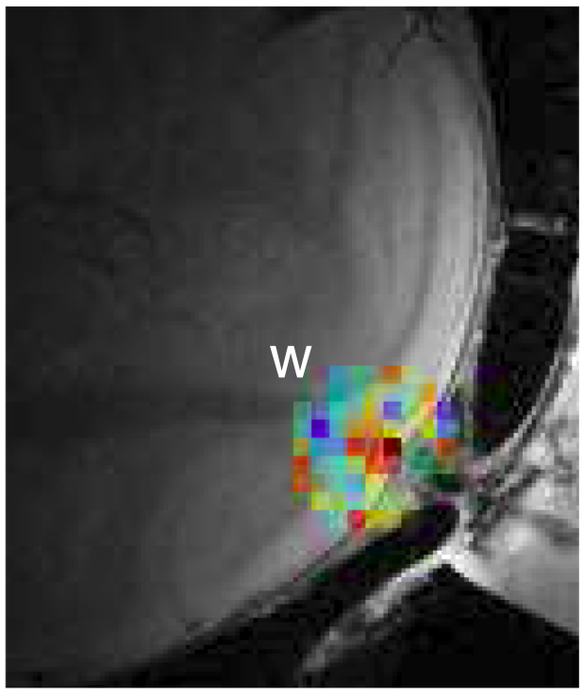
Supervised case: optimize $\mathbf{W}$ such that $\mathbf{s}(t) \approx \mathbf{y}(t)$.

Examples: linear classifiers and regression models

(SVM, LDA, lasso, OLS, Ridge regression, …)

Interpretation?

Linear projections have the same dimensionality as data, can be plotted in feature space, e.g. onto the brain.



Does this tell us what brain regions show class-specific activity?

Interpretation

Filters: tell how to weight features to extract a (target-related) component.

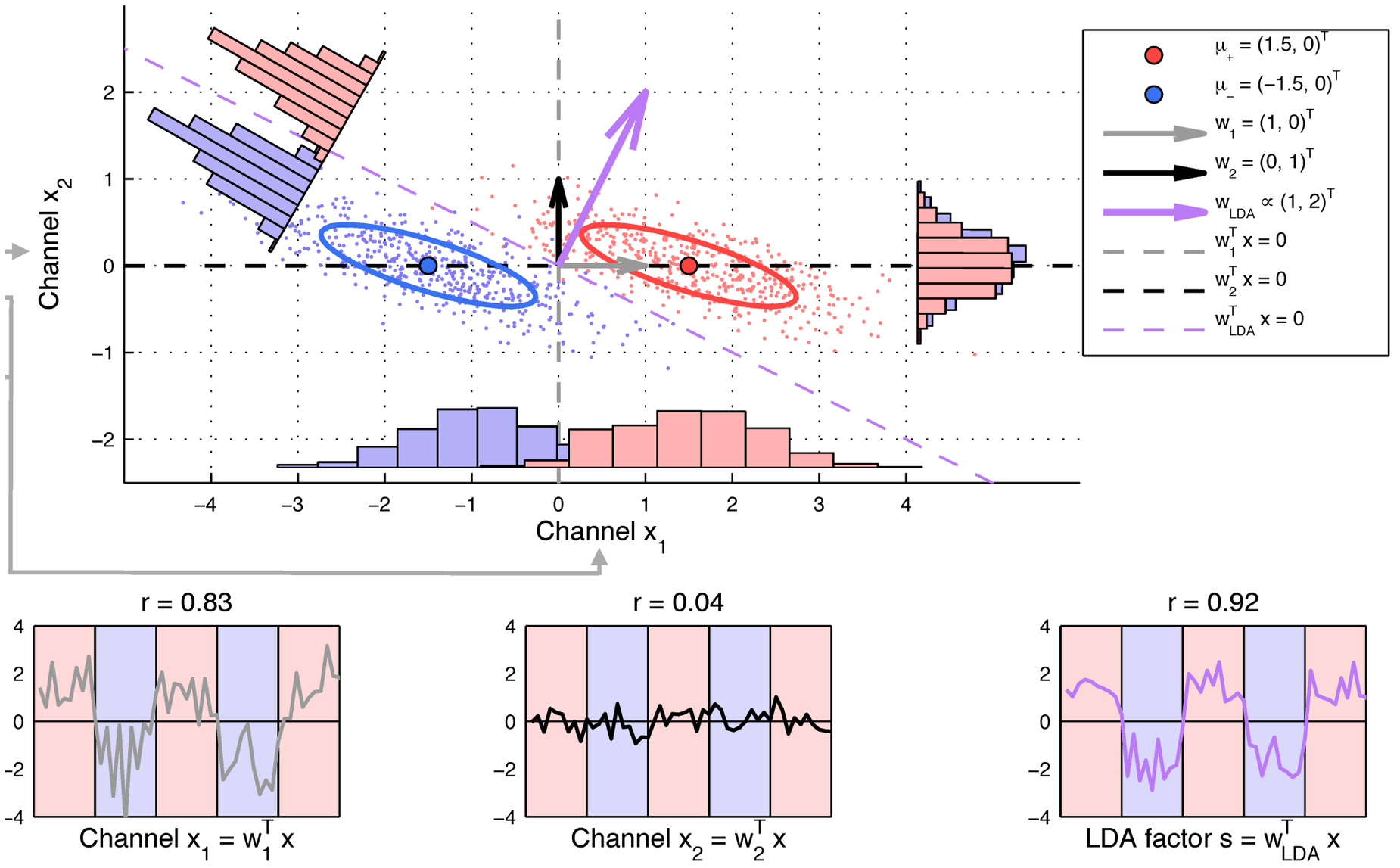
- Depend on signal and noise

→ Non-zero weight for features statically independent of the target

→ Zero weight for features related to target

→ **Not interpretable**, e.g., cannot be used to localize brain function

Example

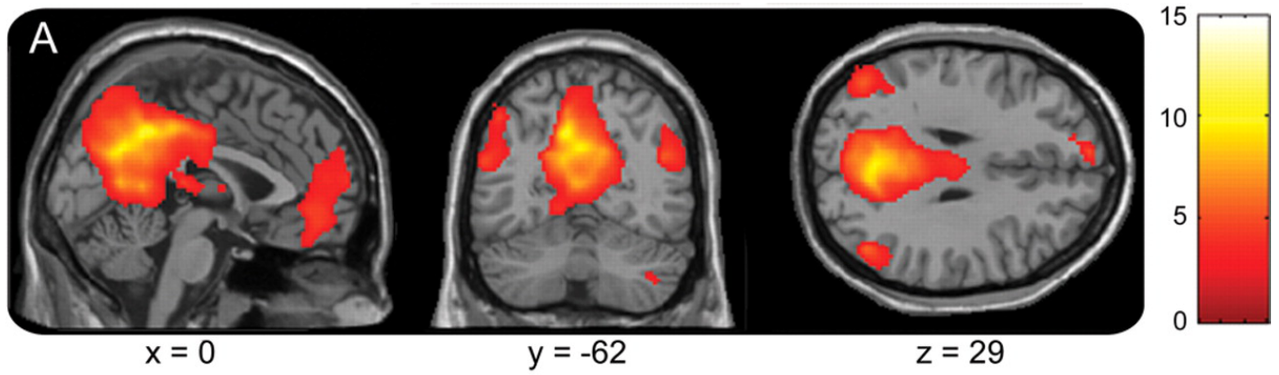


Noise correlation strongly affects $\mathbf{w}$.

Example: decoding with correlated noise

Condition-specific brain region overlapping with the ‘default-mode network’.

DMN



condition-specific

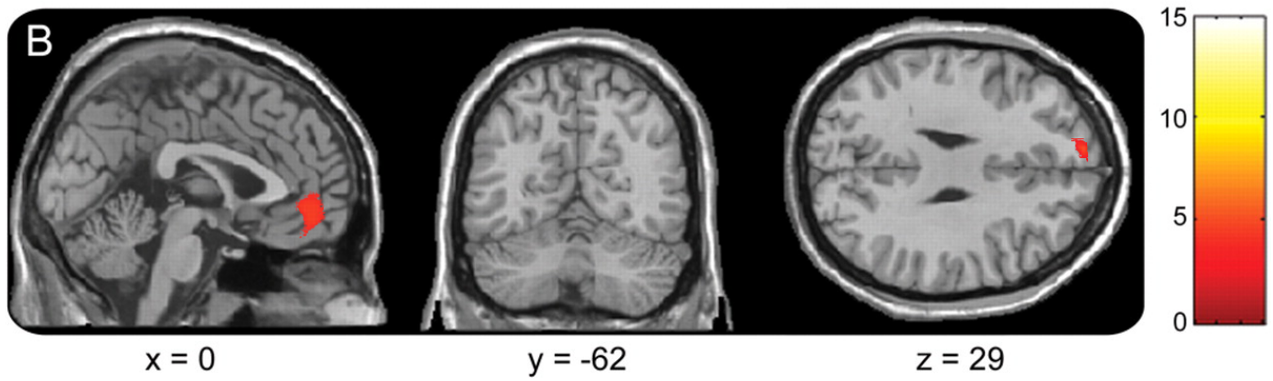


Figure from Norton et al., 2012, Neurology

Example: decoding with correlated noise

Condition-specific brain region overlapping with the ‘default-mode network’.

DMN

condition-specific

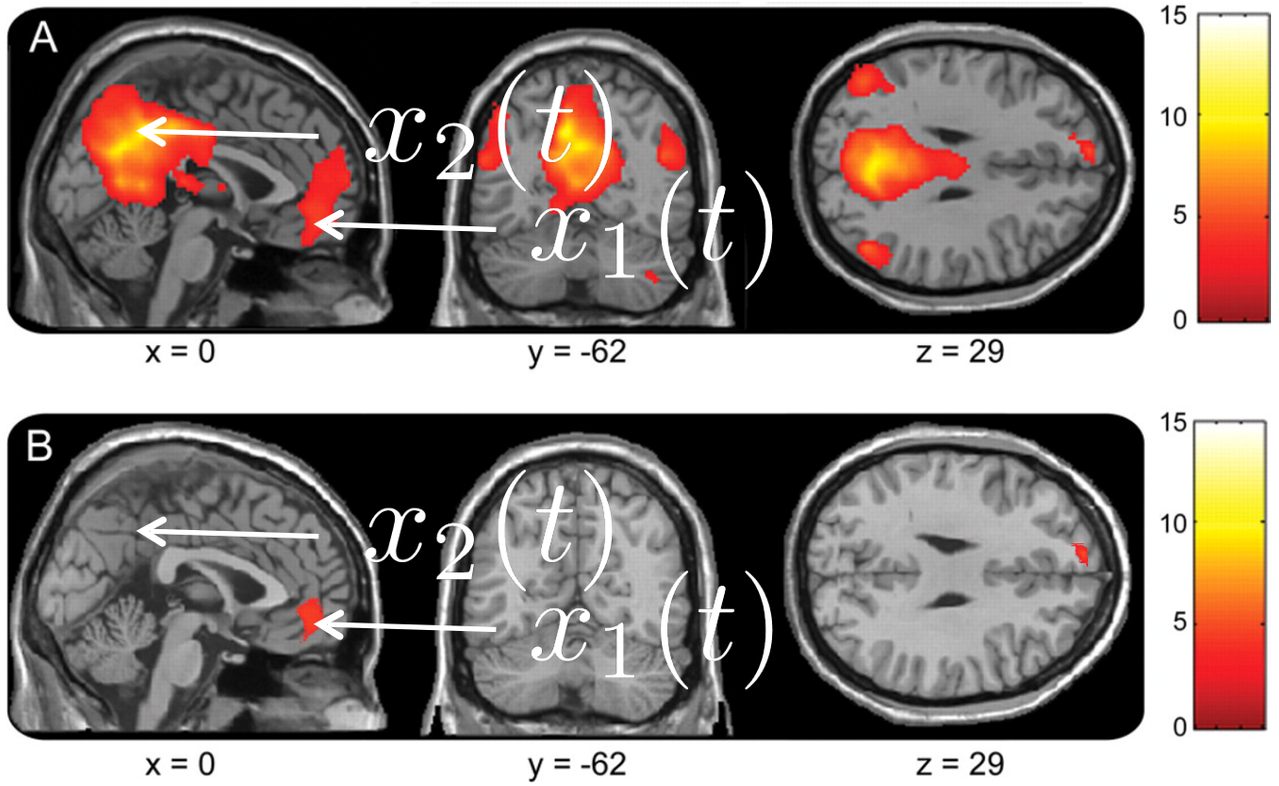


Figure from Norton et al., 2012, Neurology

Consider two voxels: $x_1(t) = y(t) + \mathrm{DMN}(t)$ and $x_2(t) = \mathrm{DMN}(t)$.

Example: decoding with correlated noise

$$x_1(t) = y(t) + \mathrm{DMN}(t) \qquad x_2(t) = \mathrm{DMN}(t)$$

The target $y(t)$ can be perfectly reconstructed by taking the difference

$$y(t) = x_1(t) - x_2(t) = \mathbf{w}^\top \mathbf{x}(t) \quad \text{with} \quad \mathbf{w} = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \text{ and } \mathbf{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}.$$

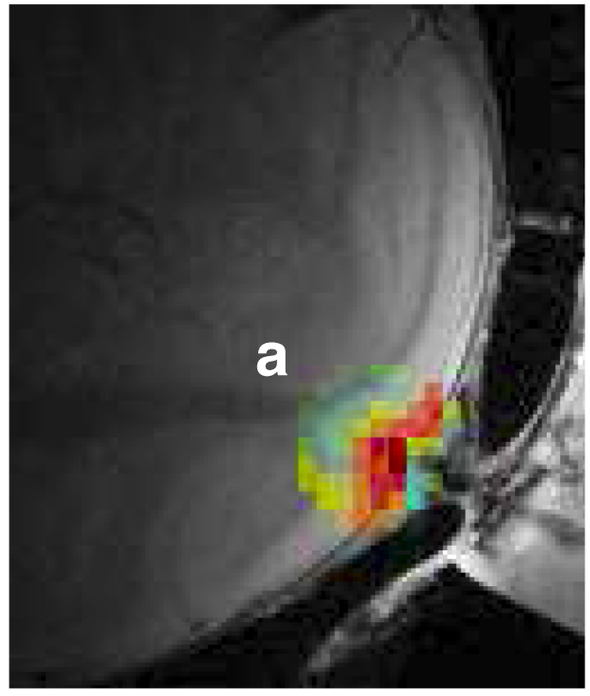
→ Despite purely measuring DMN activity, $x_2(t)$ gets a nonzero weight in $\mathbf{w}$.

Solution: forward model

Idea: express each feature as a function of the target.

$$\mathbf{x} = \mathbf{a}y + \varepsilon$$

$$x_i = a_i y + \varepsilon_i$$



The “pattern” **a** has the same dimensionality as **w**

Patterns: shows how target/underlying component is represented in each feature

→ Nonzero values only for features related to target

→ Indicate sign/strength of signal in each feature

→ **Interpretable**, e.g., can be used to localize brain activity to features

Example: decoding with correlated noise

$$x_1(t) = y(t) + \mathrm{DMN}(t) \qquad x_2(t) = \mathrm{DMN}(t)$$

The target $y(t)$ can be perfectly reconstructed by taking the difference

$$y(t) = x_1(t) - x_2(t) = \mathbf{w}^\top \mathbf{x}(t) \quad \text{with} \quad \mathbf{w} = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \text{ and } \mathbf{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}.$$

→ Despite purely measuring DMN activity, $x_2(t)$ gets a nonzero weight in $\mathbf{w}$.

The forward model is given by

$$\mathbf{x}(t) = \mathbf{a}\, y(t) + \boldsymbol{\epsilon}(t) \quad \text{with} \quad \mathbf{a} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \text{and} \quad \boldsymbol{\epsilon}(t) = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \mathrm{DMN}(t).$$

→ $x_2(t)$ gets zero weight in $\mathbf{a}$.

Can backward models be made interpretable at all?

Answer: yes, by transforming them into forward models.

backward $\quad \mathbf{W}^\top \mathbf{x}(t) = \mathbf{s}(t)$

forward $\quad \mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) + \boldsymbol{\varepsilon}(t)$

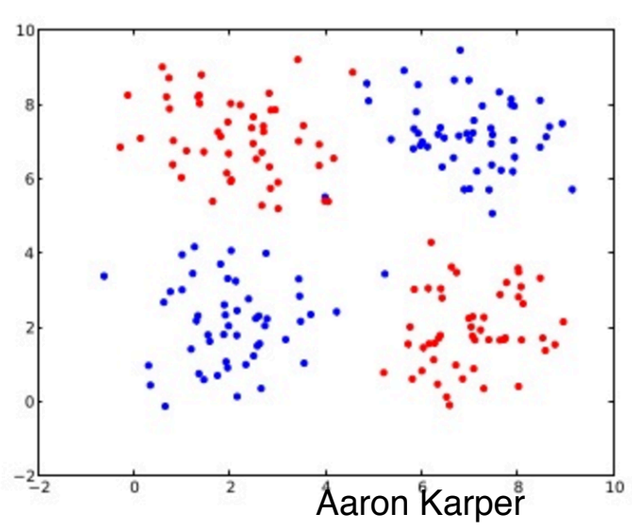
For $K = M$: $\quad \mathbf{A} = \mathbf{W}^{-\top}$.

For $K \leq M$: $\mathbf{A} = \boldsymbol{\Sigma}_\mathbf{x} \mathbf{W} \boldsymbol{\Sigma}_\mathbf{s}^{-1}$,

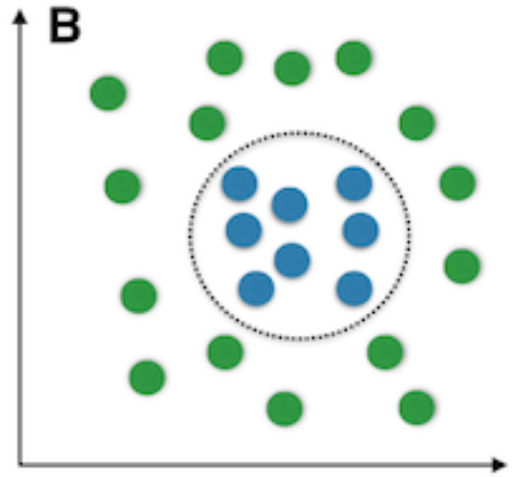
where $\boldsymbol{\Sigma}_\mathbf{x}$ and $\boldsymbol{\Sigma}_\mathbf{s}$ are the covariance matrices of $\mathbf{x}(t)$ and $\mathbf{s}(t)$.

For $K = 1$ or uncorrelated components : $\mathbf{A} \propto \boldsymbol{\Sigma}_\mathbf{x} \mathbf{W}$.

Non-linear data



Aaron Karper



Sebastian Raschka

Solutions

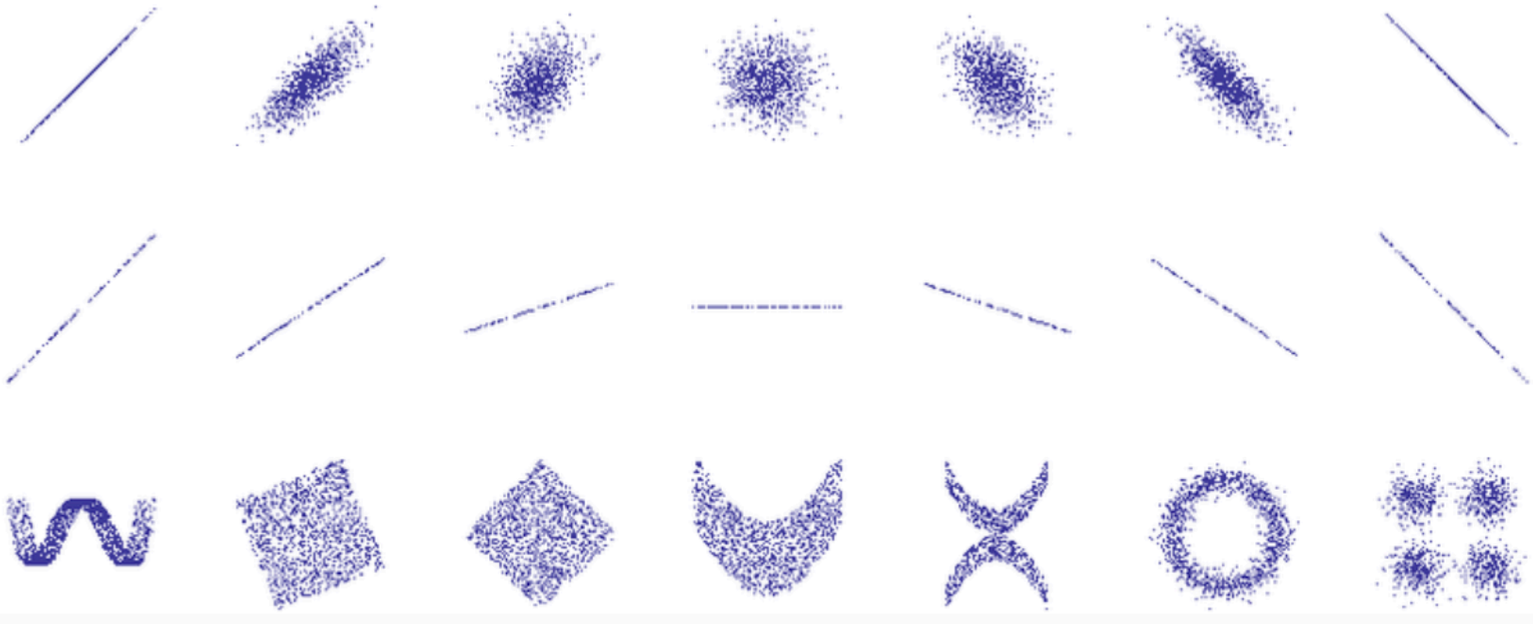
- Non-linear classifiers/decision boundaries
- Non-linear features

Covariance and correlation

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

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

Correlation measures the linear relationship between X and Y:

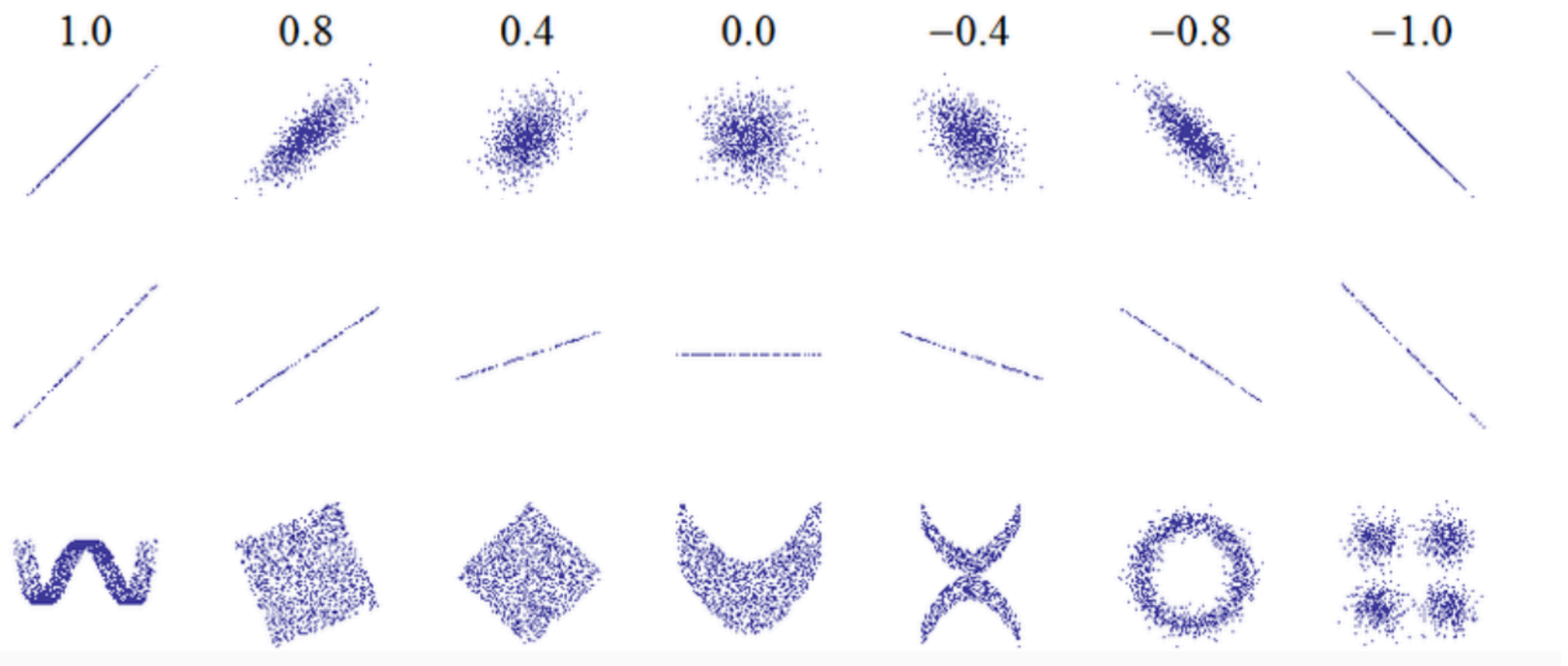


Covariance and correlation

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

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

Correlation measures the linear relationship between X and Y:

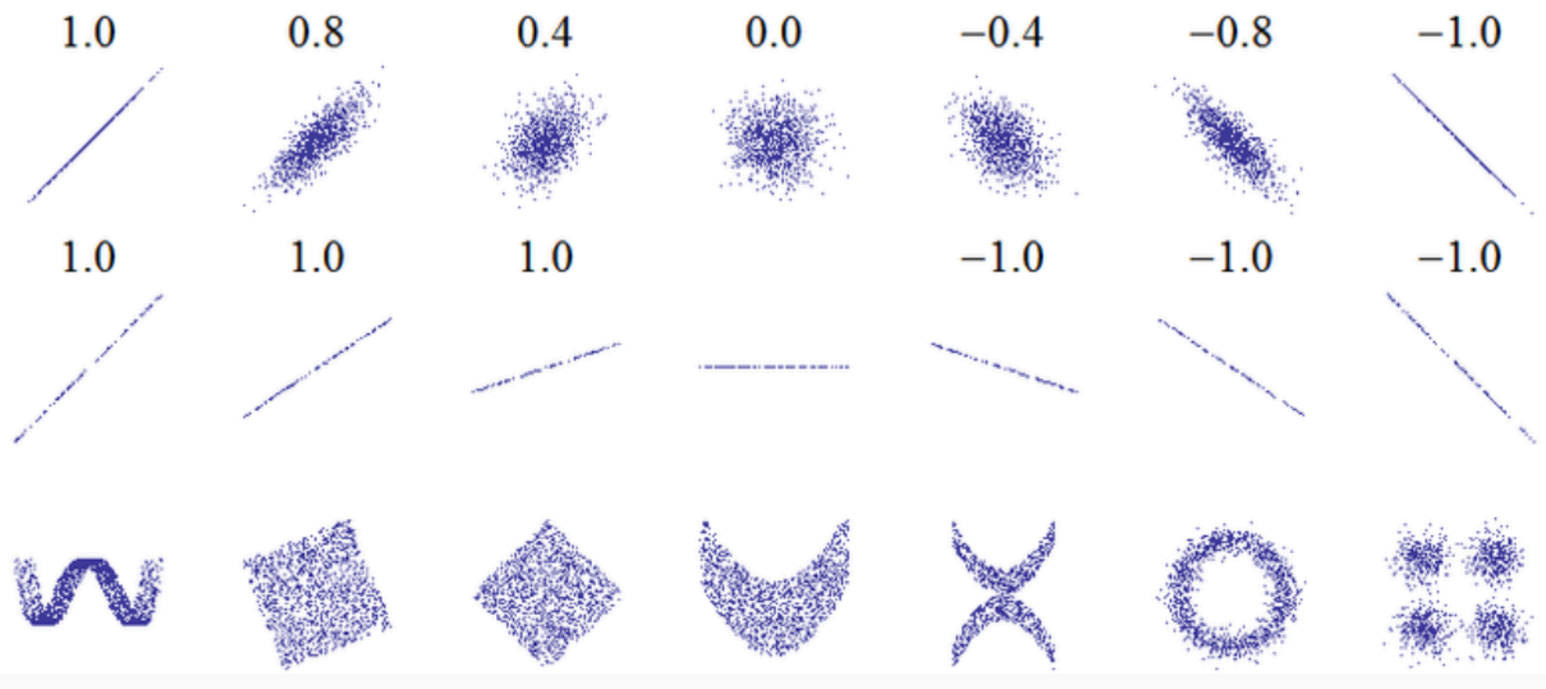


Covariance and correlation

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

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

Correlation measures the linear relationship between X and Y:

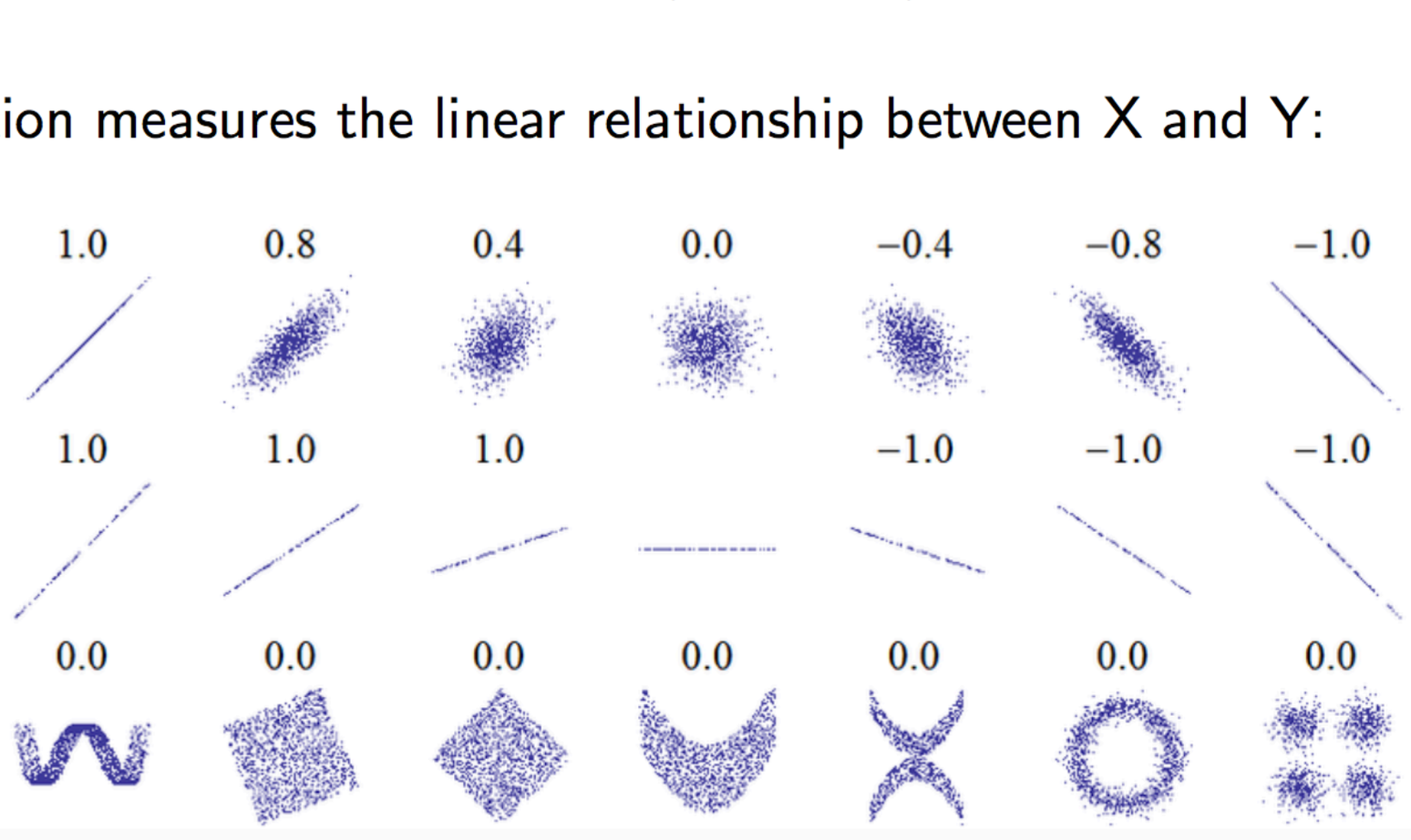


Covariance and correlation

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

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

Correlation measures the linear relationship between X and Y:



Mini-exercise

1. Consider a random variable X, $\alpha \in \mathbb{R} \setminus \{0\}$ and $Y = \alpha X$. Then:

   (A) $\text{Corr}(X, Y) = \alpha$
   (B) $\text{Corr}(X, Y) = \text{sign}(\alpha)$
   (C) $\text{Corr}(X, Y) = 1$

Mini-exercise

1. Consider a random variable X, $\alpha \in \mathbb{R} \setminus \{0\}$ and $Y = \alpha X$. Then:

   (A) $\text{Corr}(X, Y) = \alpha$
   (B) $\text{Corr}(X, Y) = \text{sign}(\alpha)$
   (C) $\text{Corr}(X, Y) = 1$

2. Consider two random variables X and Y with covariance matrix $\begin{pmatrix} 4 & 0.5 \\ 0.5 & 1 \end{pmatrix}$. Then $\text{Corr}(X, Y) = ?$

Mini-exercise

1. Consider a random variable X, $\alpha \in \mathbb{R} \setminus \{0\}$ and $Y = \alpha X$. Then:

   (A) $\mathrm{Corr}(X, Y) = \alpha$
   (B) $\mathrm{Corr}(X, Y) = \mathrm{sign}(\alpha)$
   (C) $\mathrm{Corr}(X, Y) = 1$

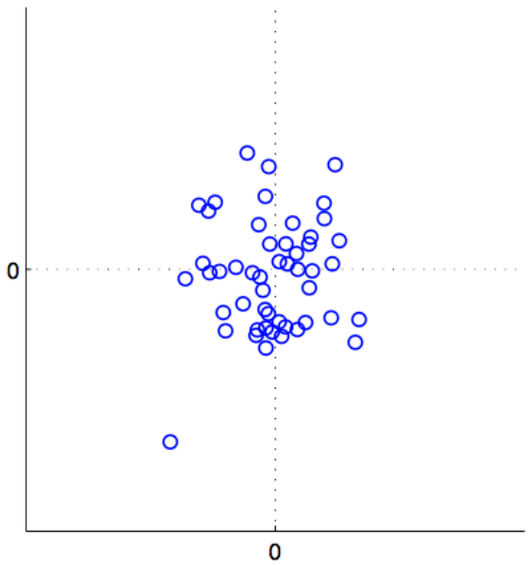
2. Consider two random variables X and Y with covariance matrix $\begin{pmatrix} 4 & 0.5 \\ 0.5 & 1 \end{pmatrix}$. Then $\mathrm{Corr}(X, Y) = ?$

3. Consider a data set with two data points: $\begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} -1 \\ -1 \end{bmatrix}$. Compute the covariance matrix.

Correlated data and linear mappings

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

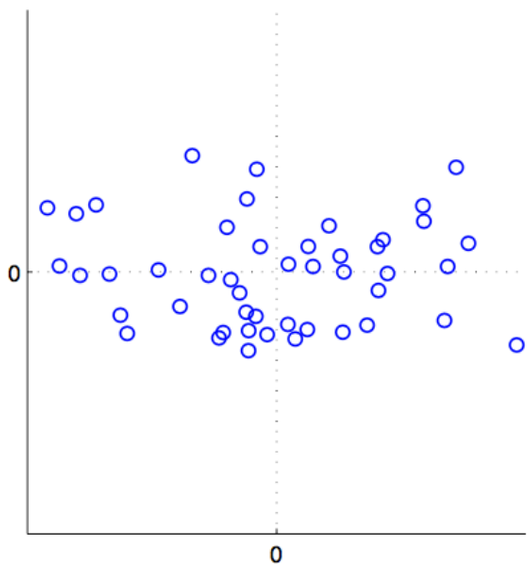
Uncorrelated



$$X \sim \mathcal{N}(\mathbf{0}, I)$$

$$\text{Cov}(X) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

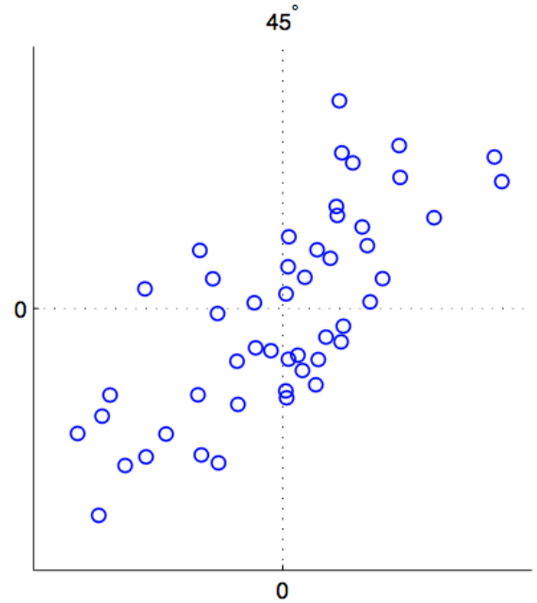
Uncorrelated, scaled



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

$$\text{Cov}(X) = \begin{bmatrix} 9 & 0 \\ 0 & 1 \end{bmatrix}$$

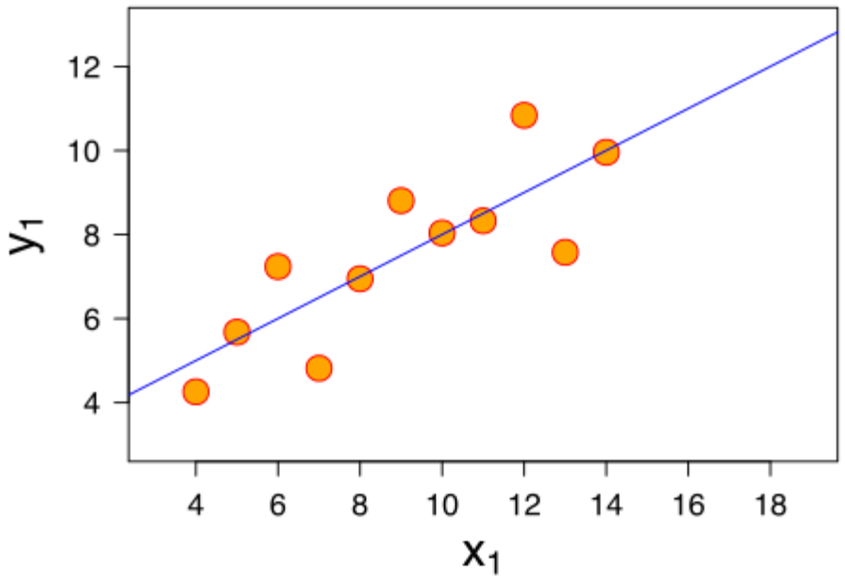
Scaled, rotated by 45°, $\phi = \pi/4$

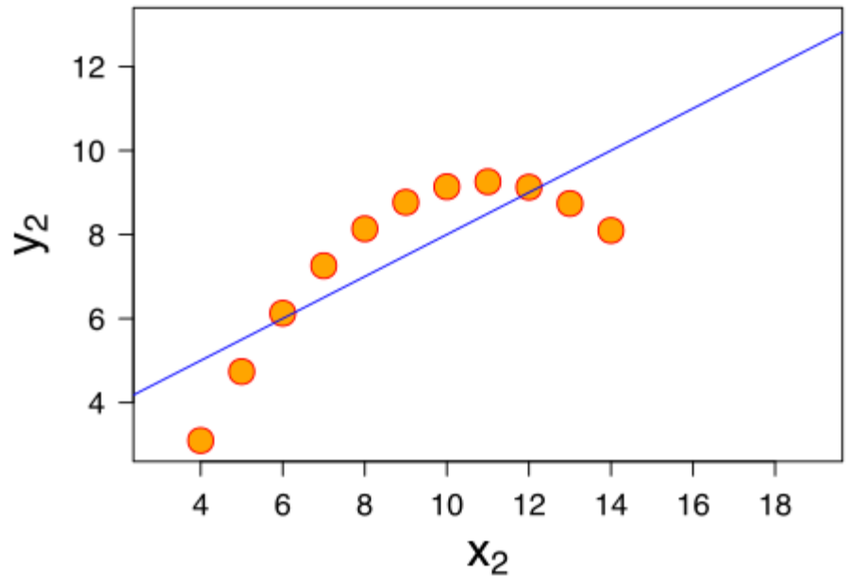


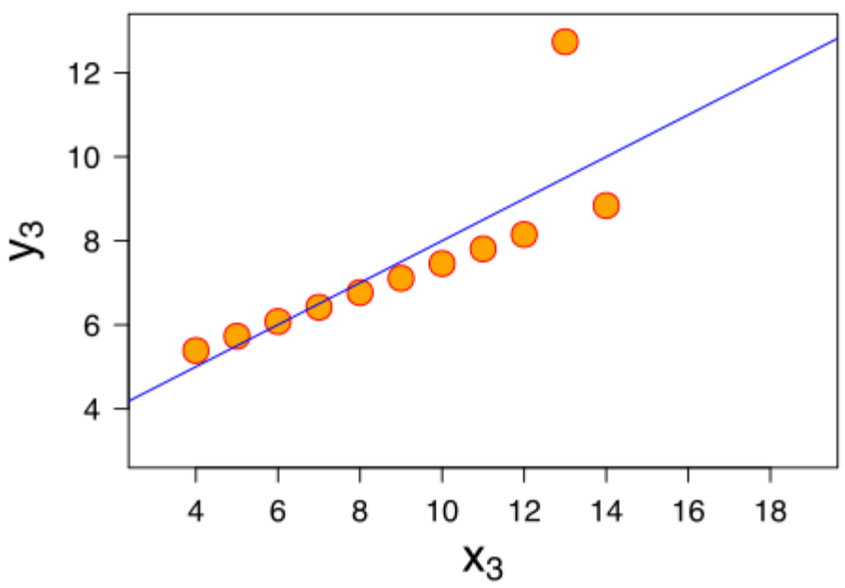
$$\begin{bmatrix} cos(\phi) & -sin(\phi) \\ sin(\phi) & cos(\phi) \end{bmatrix} \begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix} X$$

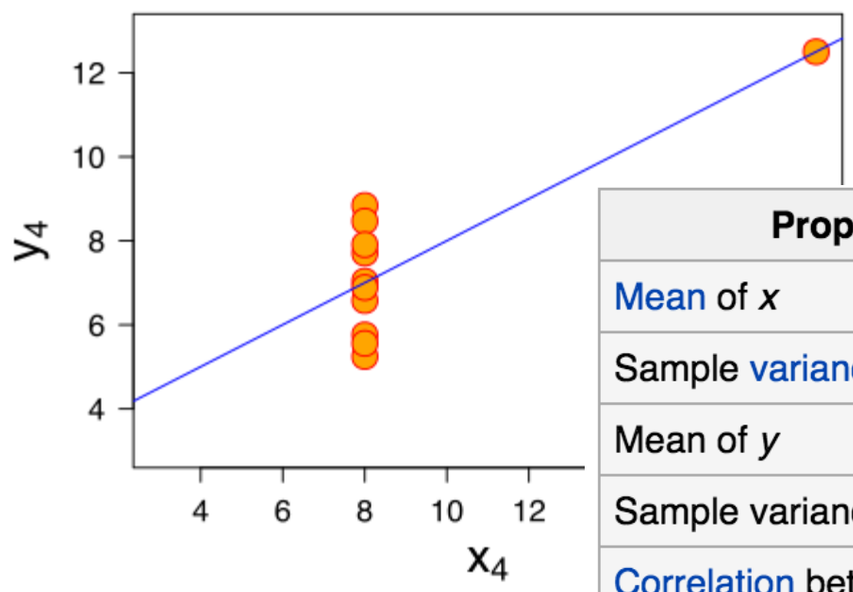
$$\text{Cov}(X) = \begin{bmatrix} 5 & 4 \\ 4 & 5 \end{bmatrix}$$

Anscomb's quartet









Property	Value
Mean of *x*	9
Sample variance of *x*	11
Mean of *y*	7.50
Sample variance of *y*	4.125
Correlation between *x* and *y*	0.816
Linear regression line	*y* = 3.00 + 0.500*x*

Summary

- Discriminative modeling: estimate posterior or decision function directly
- Nearest-centroid classifier suboptimal if features are correlated
- Linear discriminant based on Fisher's criterion leads to Bayes-optimal solution if classes are Gaussian with equal covariance
- Classifier weights tell us what features are important for classification, but not what features actually differ between classes
- Many applications
- Multiple classes, non-linear features, large-margin classifiers