Analiza wariancji i metody klasyfikacyjne
Analysis of variance and classification methods

Classification

lecture 7

9 December 2019

Ilona Anna Urbaniak (PK)

Marcin Wolter (IFJ PAN)

e-mail: marcin.wolter@ifj.edu.pl, phone: 12 662 8024

Slides: https://indico.ifj.edu.pl/event/271/

Github repository

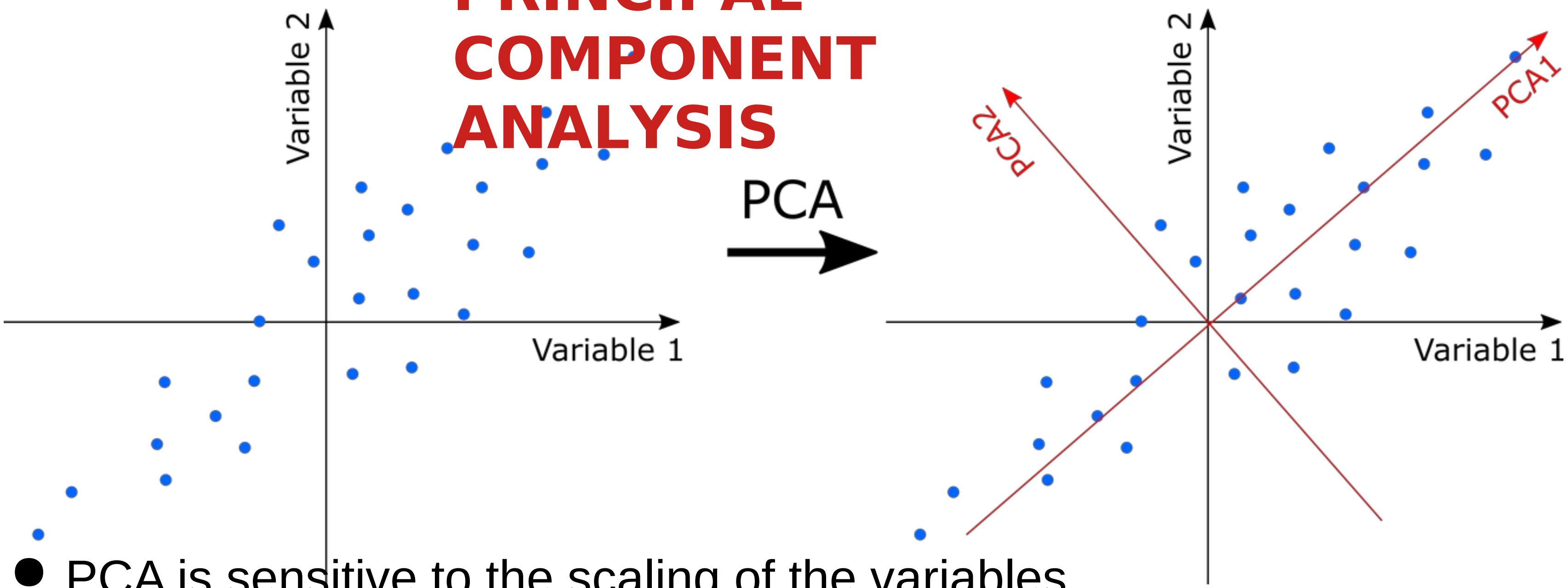
https://github.com/marcinwolter/ANOVA_2019

Some python examples (just a reminder)

Last lecture - PCA

- Suppose we have a population measured on p random variables $X_1,\ldots,X_p$.
- Goal: a new set of p axes (linear combinations of the original p axes) in the directions of greatest variability:

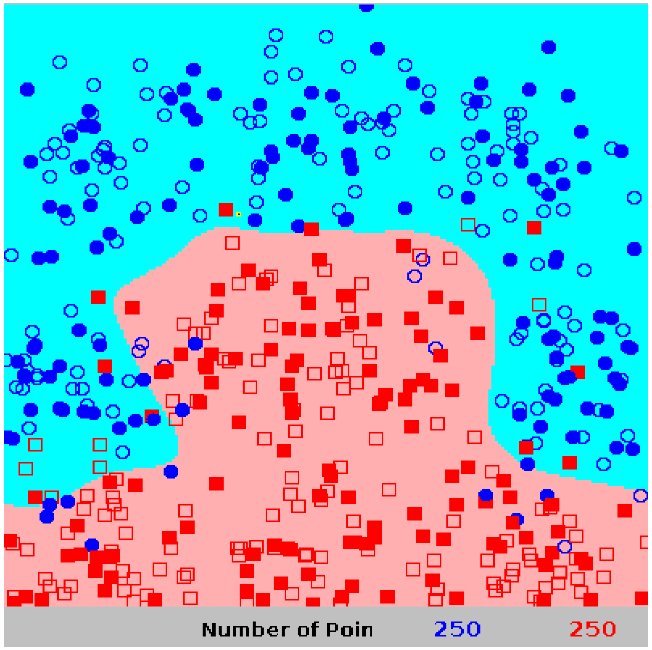
PRINCIPAL COMPONENT ANALYSIS



- PCA is sensitive to the scaling of the variables.

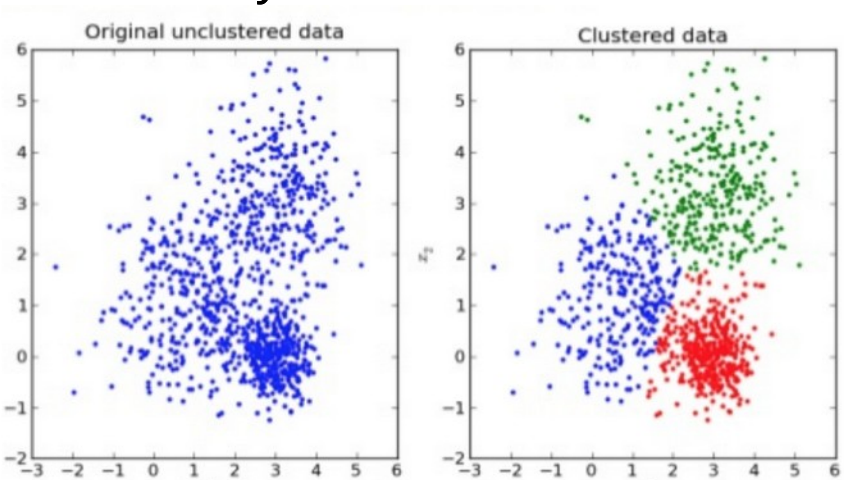
Classification

- In statistics, classification is the problem of identifying to which of a set of categories a new observation belongs, on the basis of a training set of data containing observations whose category membership is known.
- Classification is an example of pattern recognition.
- Example: assigning a given email to the "spam" or "non-spam" class, and assigning a diagnosis to a given patient based on observed characteristics of the patient (sex, blood pressure, presence or absence of certain symptoms, etc.).



Classification is a part of Machine Learning

- **Machine learning** is a field of computer science that gives computer systems the ability to "learn" (i.e. progressively improve performance on a specific task) with data, without being explicitly programmed.
- Problems:
  - Supervised learning (classification & regression)
  - Clustering (unsupervised learning)
  - Dimensionality reduction
  - Reinforcement learning
  - Many others….



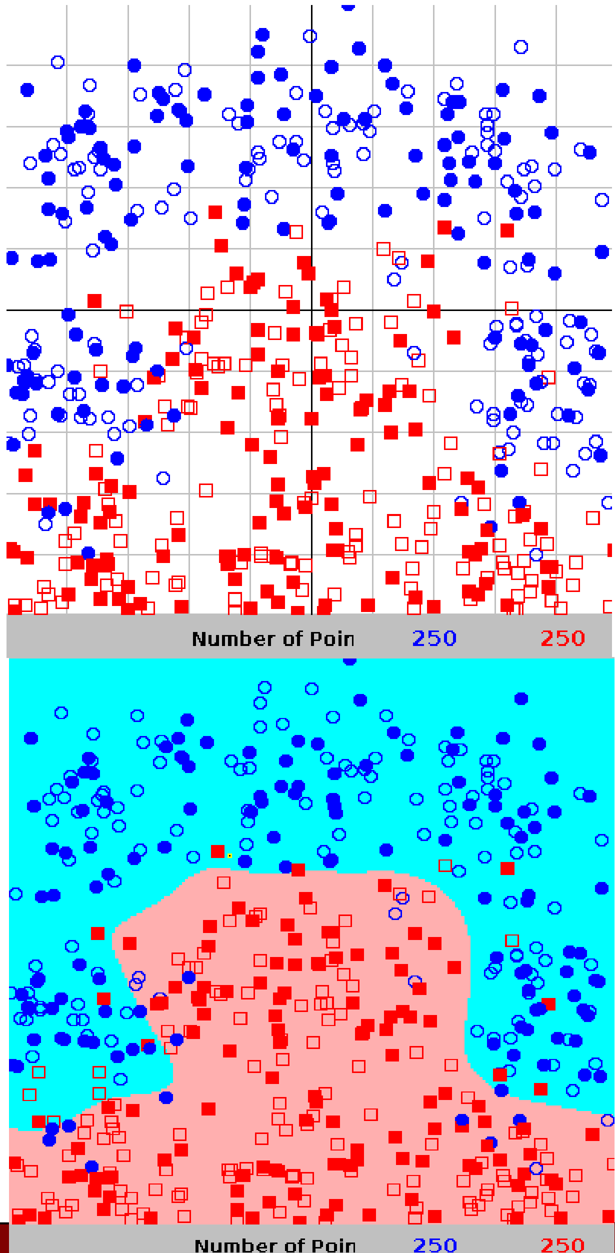
➢Unsupervised Learning

❑Technique of trying to find hidden structure in unlabeled data

➢Supervise Learning

❑Technique for creating a function from training data. The training data consist of pairs of input objects (typically vectors), and desired outputs.

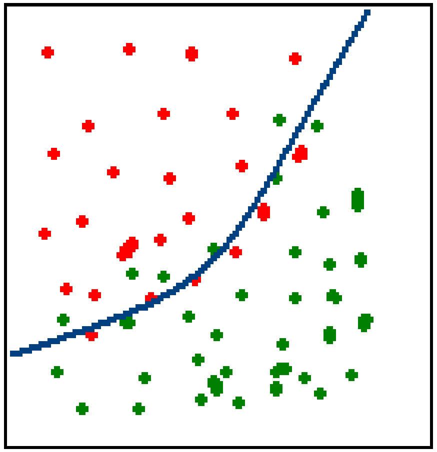
How do the (supervised) machine learning algorithms work?



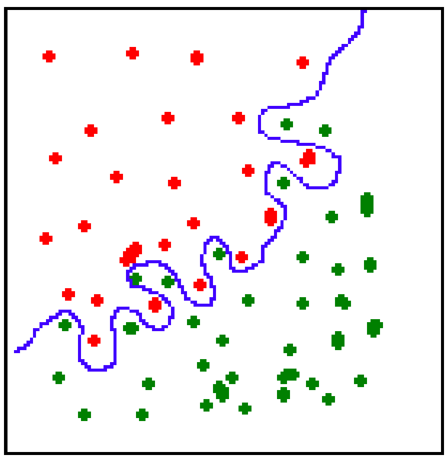
- We need **training data**, for which we know the correct answer, whether it's a signal or background. We divide the data into two samples: training and test.
- We find the best function ***f(x)*** which describes the probability, that a given event belongs to the class "signal". This is done by minimizing the loss function (for example χ^2).
- Different algorithms differ by: the class of function used as ***f(x)*** (linear, non-linear etc), loss function and the way it's minimized.
- All these algorithms try to approximate the unknown *Bayessian Decisive Function* (BDF) relying on the finit training sample.

BDF -an ideal classification function given by the unknown probability densities of signal and background.

Overtraining

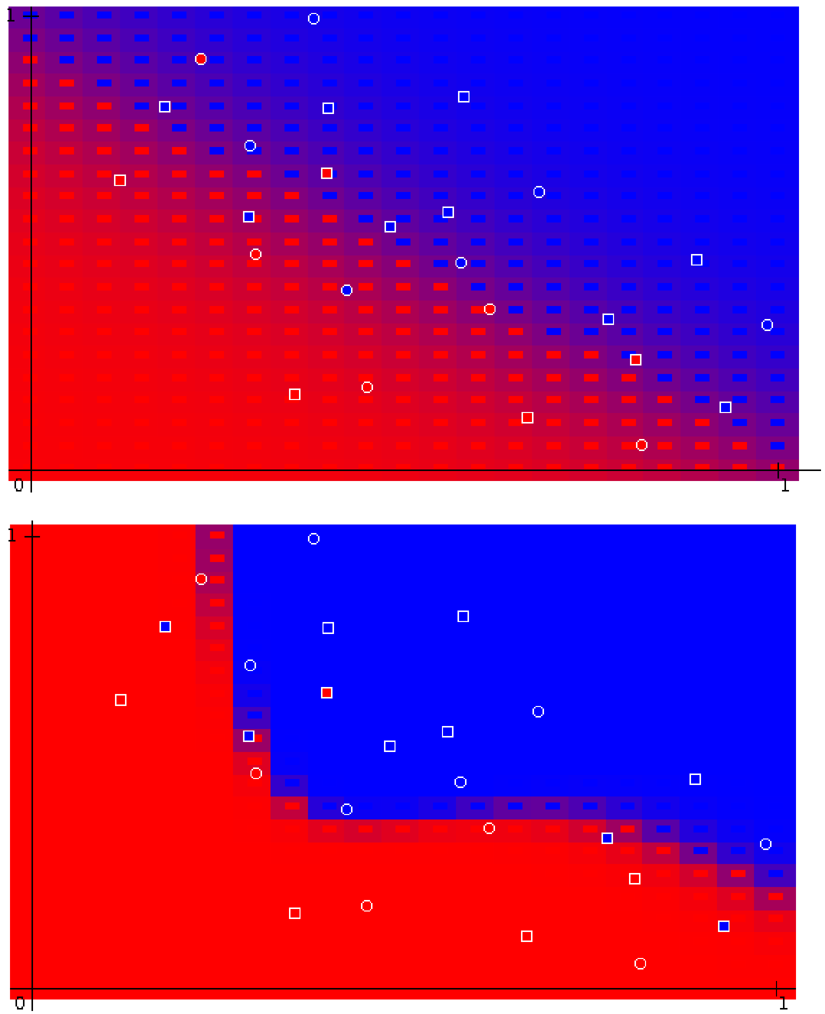


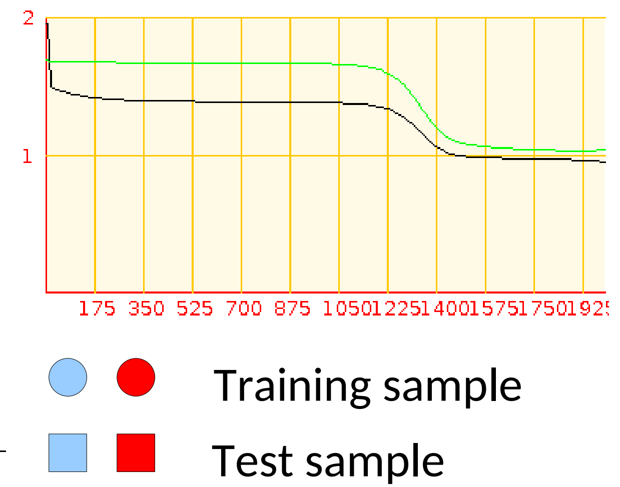
Correct

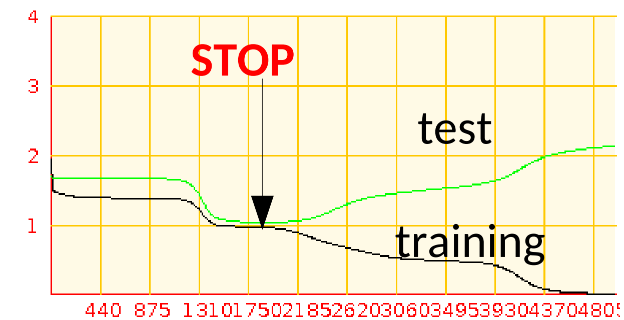


Overtraining

- **Overtraining** – algorithm “learns” the particular events, not the rules.
- This effect important for all ML algorithms.
- Remedy – checking with another, independent dataset.







Example of using Neural Network.

Classification

A Bayes classifier (optimal classifier):

$$p(S|x)=\frac{p(x|S)\,p(S)}{p(x|S)\,p(S)+p(x|B)\,p(B)}$$

where ***S*** is associated with y = **1** and ***B*** with y = **0**. **Bayes classifier** *accepts events x if p*(***S***|*x*) **>** ***cut*** *as belonging to* ***S***.

We need to approximate probability distributions *P(**x**|**S**)* and *P(**x**|**B**).*

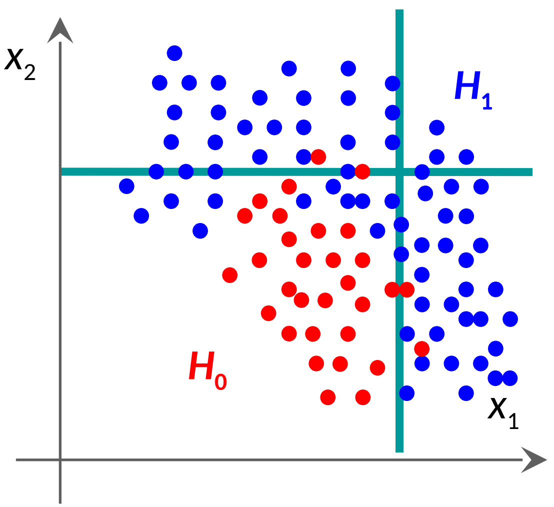
- *If your goal is to **classify objects** with the fewest errors, then the **Bayes classifier** is the **optimal** solution.*
- Consequently, if you have a classifier known to be ***close*** to the **Bayes limit**, then *any* other classifier, *however sophisticated*, can ***at best*** be only marginally better than the one you have.
  - =>If your problem is **linear** you don't gain anything by using sophisticated **Neural Network**
- ***All* classification methods, such as all we will be talking about, are different numerical approximations of the Bayes classifier.**

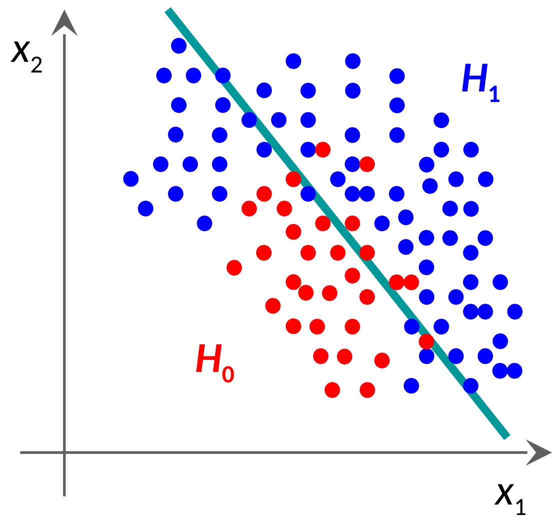
Types of algorithms

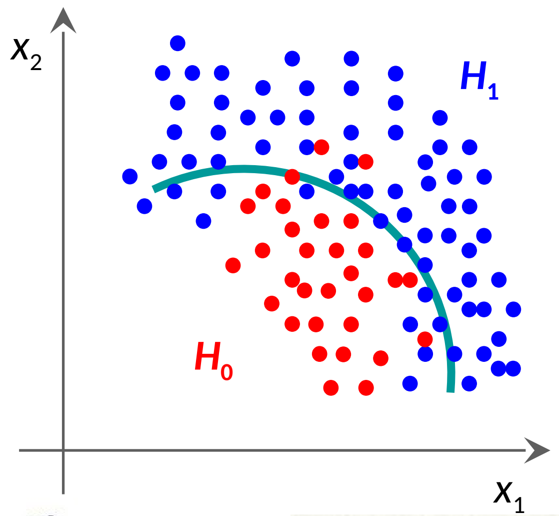
How to use the information available

Classification: find a function ***f(x1,x2)*** giving the probability, that a given data point belongs to a given class (signal vs background).

Simple cuts (easy and intuitive)	Linear (fast and stable)	Non-linear (most effective)



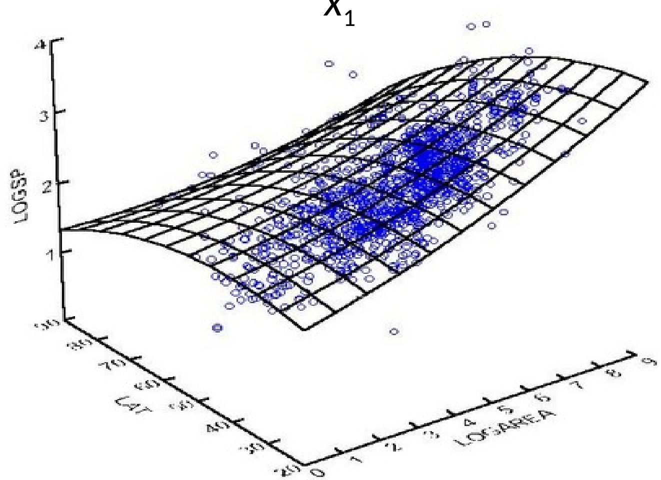


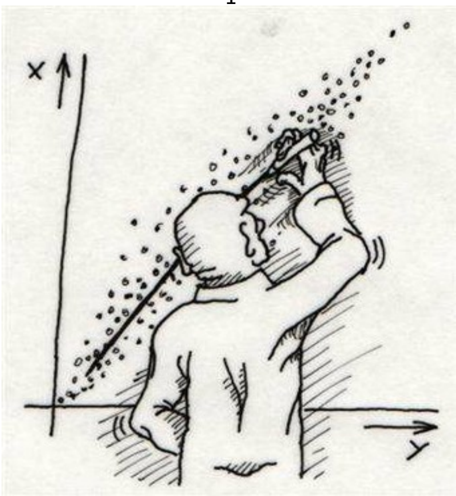


Regression: fit a continuous function

(find particle energy from the detector

readouts).





Bayes theorem

Bayes' theorem is stated mathematically as the following equation:

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

where **A** and **B** are events and **P(B)≠ 0**.

P(A | B) is a conditional probability: the likelihood of event **A** occurring given that **B** is true.
P(B | A) is also a conditional probability: the likelihood of event **B** occurring given that **A** is true.
P(A) and **P(B)** are the probabilities of observing **A** and **B** independently of each other; this is known as the marginal or unconditional probability.

Bayes decision theory

- Statistical nature of feature vectors

$$\mathbf{x} = \left[x_1, x_2, \ldots, x_l \right]^T$$

- Assign the pattern represented by feature vector χ to the most probable of the available classes

$$\omega_1, \omega_2, \ldots, \omega_M$$

That is $\mathbf{x} \rightarrow \omega_i : P\left(\omega_i \middle| \mathbf{x} \right)$

maximum

14

Bayes Decision Theory

- Computation of a-posteriori probabilities
  - Assume known
    - a-priori probabilities

$$P(\omega_1), P(\omega_2)..., P(\omega_M)$$

- $$p(\boldsymbol{x}|\omega_i), i=1,2,\ldots,M$$

This is also known as the likelihood of $\boldsymbol{x}$ $w.r.$ to ω_i.

15

- The Bayes rule (M=2)

$$p(\boldsymbol{x})P(\omega_i|\boldsymbol{x})=p(\boldsymbol{x}|\omega_i)P(\omega_i)\Rightarrow$$

$$P(\omega_i|\boldsymbol{x})=\frac{p(\boldsymbol{x}|\omega_i)P(\omega_i)}{p(\boldsymbol{x})}$$

where

$$p(\boldsymbol{x})=\sum_{i=1}^{2}p(\boldsymbol{x}|\omega_i)P(\omega_i)$$

The Bayes classification rule (for two classes M=2)

- Given x classify it according to the rule

$$\boxed{\begin{array}{ll} \text{If } P(\omega_1|x) > P(\omega_2|\underline{x}) & x \to \omega_1 \\ \text{If } P(\omega_2|x) > P(\omega_1|x) & x \to \omega_2 \end{array}}$$

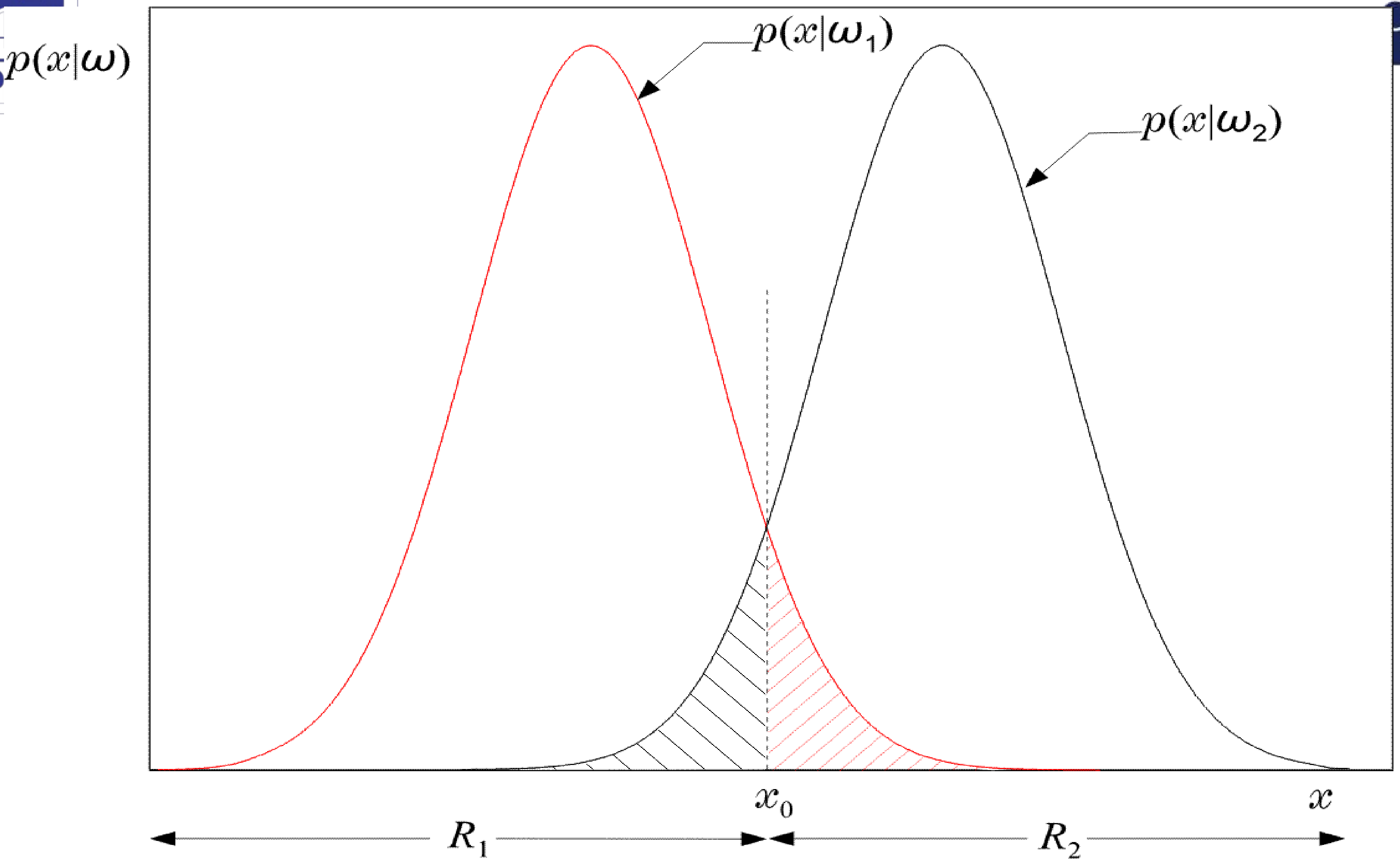
- Equivalently: classify x according to the Bayes rule

$$\boxed{p(x|\omega_1)P(\omega_1)(><)p(x|\omega_2)P(\omega_2)}$$

- For equiprobable classes the test becomes

$$\boxed{p(x|\omega_1)(><)p(x|\omega_2)}$$

17



$$R_1(\to\ \omega_1)\ \text{and}\ R_2(\to\ \omega_2)$$

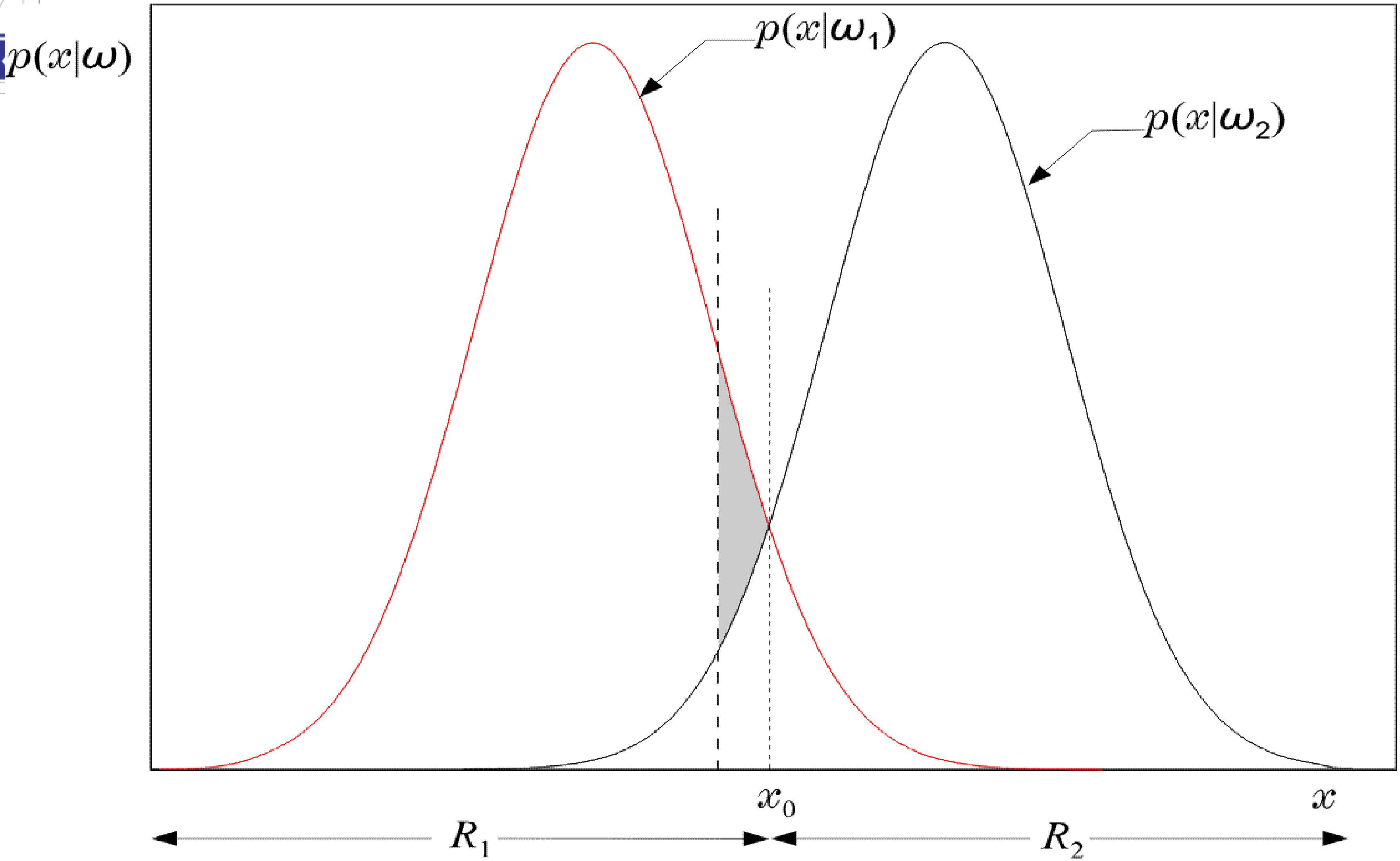
18

- Equivalently in words: Divide space in two regions

$$\boxed{\begin{array}{l}\text{If } \boldsymbol{x} \in R_1 \Rightarrow \underline{x} \text{ in } \omega_1 \\ \text{If } \boldsymbol{x} \in R_2 \Rightarrow \underline{x} \text{ in } \omega_2\end{array}}$$

- Probability of error
  - Total shaded area
  - $$P_e = \frac{1}{2}\int_{-\infty}^{x_0} p(x|\omega_2)\,dx + \frac{1}{2}\int_{x_0}^{+\infty} p(x|\omega_1)\,dx$$

❖ **Bayesian classifier is OPTIMAL with respect to minimizing the classification error probability!!!!**



– Indeed: Moving the threshold the total shaded area INCREASES by the extra “grey” area.

Classification Accuracy and Error

- Classification accuracy is the ratio of correct predictions to total predictions made.
  - **classification accuracy = correct predictions / total predictions**
- It is often presented as a percentage by multiplying the result by 100.
  - **classification accuracy = correct predictions / total predictions * 100**
- Classification accuracy can also easily be turned into a misclassification rate or error rate by inverting the value, such as:
  - **error rate = (1 - (correct predictions / total predictions)) * 100**

Confusion matrix

- Example confusion matrix (recognition of dogs vs. cats)

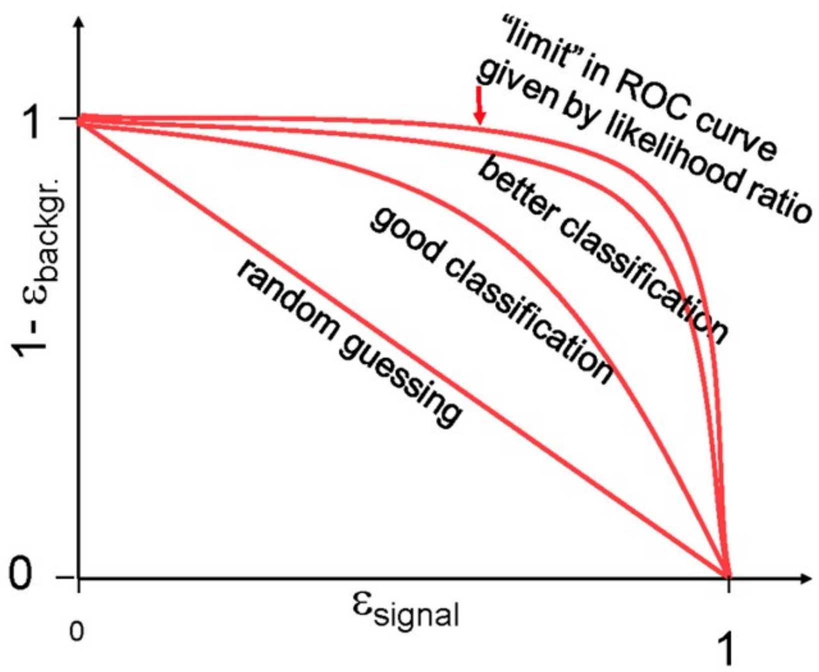
		Actual class	
		Cat	Dog
Predicted class	Cat	**5**	2
	Dog	3	**3**

Confusion Matrix

<table>
<tr><td></td><td></td><td colspan="2">True condition</td></tr>
<tr><td></td><td>Total population</td><td>Condition positive</td><td>Condition negative</td></tr>
<tr><td rowspan="2">Predicted condition</td><td>Predicted condition positive</td><td>True positive</td><td>False positive,
Type I error</td></tr>
<tr><td>Predicted condition negative</td><td>False negative,
Type II error</td><td>True negative</td></tr>
<tr><td></td><td></td><td>True positive rate (TPR), Recall, Sensitivity, probability of detection, Power
$= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$</td><td>False positive rate (FPR), Fall-out, probability of false alarm
$= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$</td></tr>
<tr><td></td><td></td><td>False negative rate (FNR), Miss rate
$= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$</td><td>Specificity (SPC), Selectivity, True negative rate (TNR)
$= \frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$</td></tr>
</table>

ROC curve

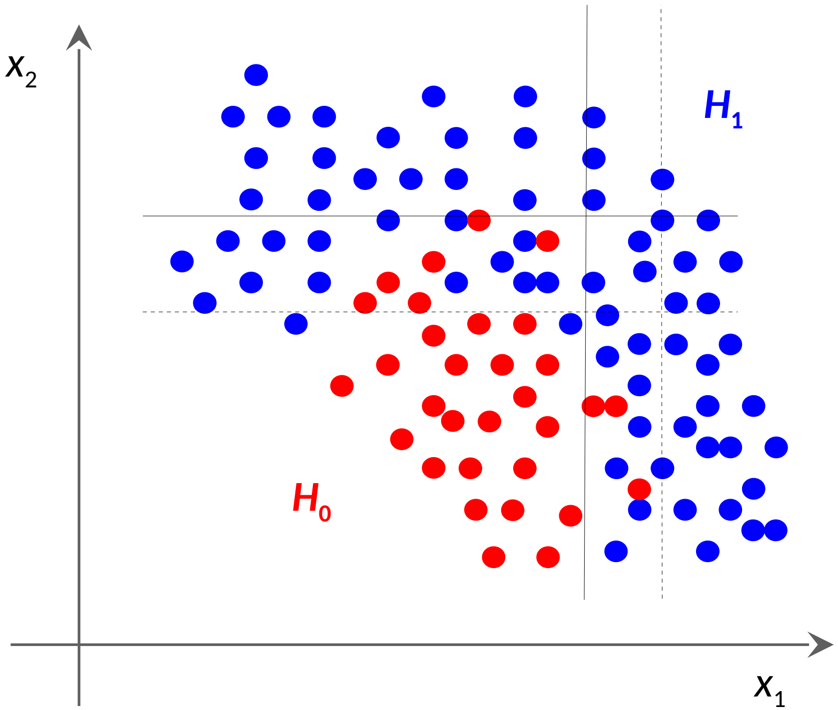
- ROC (Receiver Operation Characteristic) curve was first used to calibrate radars.
- Two class classification.
- Shows the background rejection ($1-\varepsilon_B$) vs. signal efficiency ε_B. Shows how good the classifier is.
- The integral of ROC could be a measure of the classifier quality:



Integral(ROC) = ½ – random

Integral(ROC) = 1 - ideal

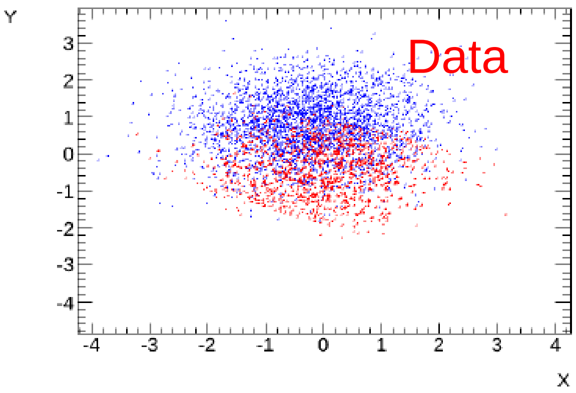
Cuts

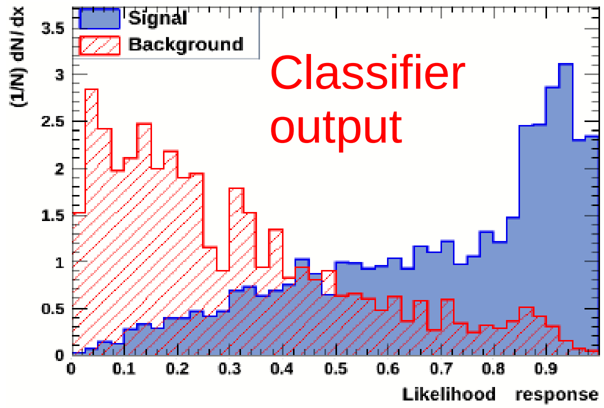


Optimization of cuts:

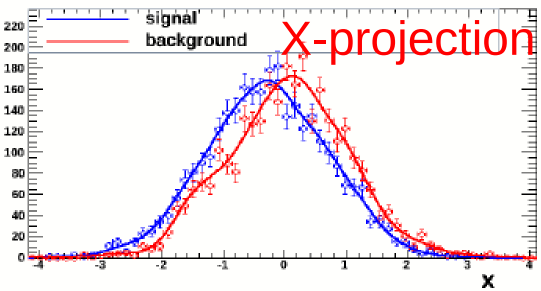
- Move cuts as long as we get the optimal signal vs. background selection. For a given signal efficiency we find the best background rejection → we get the entire ROC curve.
- Optimization methods:
  - Brute force
  - Genetic algorithms
  - Many others...

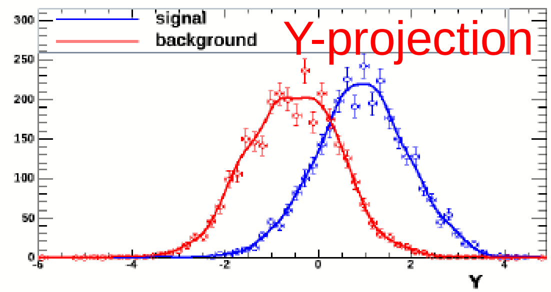
Naive Bayes classifier





Frequently also called **“projected likelihood".**





- Based on the assumption, that variables are independent (so „naive"):

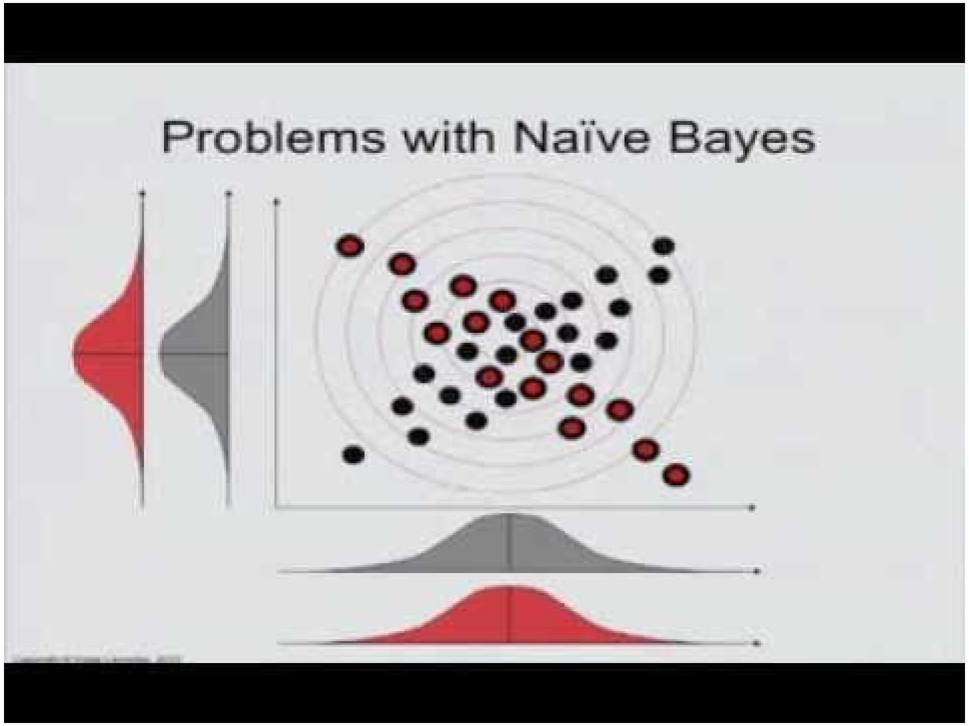
$$P(y \mid x_1, \dots, x_n) = \frac{P(y) P(x_1, \dots x_n \mid y)}{P(x_1, \dots, x_n)}$$

“Naive” assumption:

$$P(x_i | y, x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) = P(x_i | y),$$

Naive Bayes

- Output probability is **a product of probabilities for all variables.**
- Fast and stable
- It turns out that the Naive – Bayes classifier works reasonably well even in cases that violate the independence assumption.



In most real-life cases NB is suboptimal, sometimes it might fail.

K-Nearest Neighbors

- These classifiers are *memory-based* and require no model to be fit.
- Training data: $(g_i, x_i), i = 1, 2, \ldots, N$

  - Define distance on input *x* (e.g. Euclidian distance)

  - Classify new instance by looking at the label of the single closest sample in the training set:

$$\hat{G}(x^*) = argmin_i d(x_i, x^*)$$

K-Nearest Neighbors

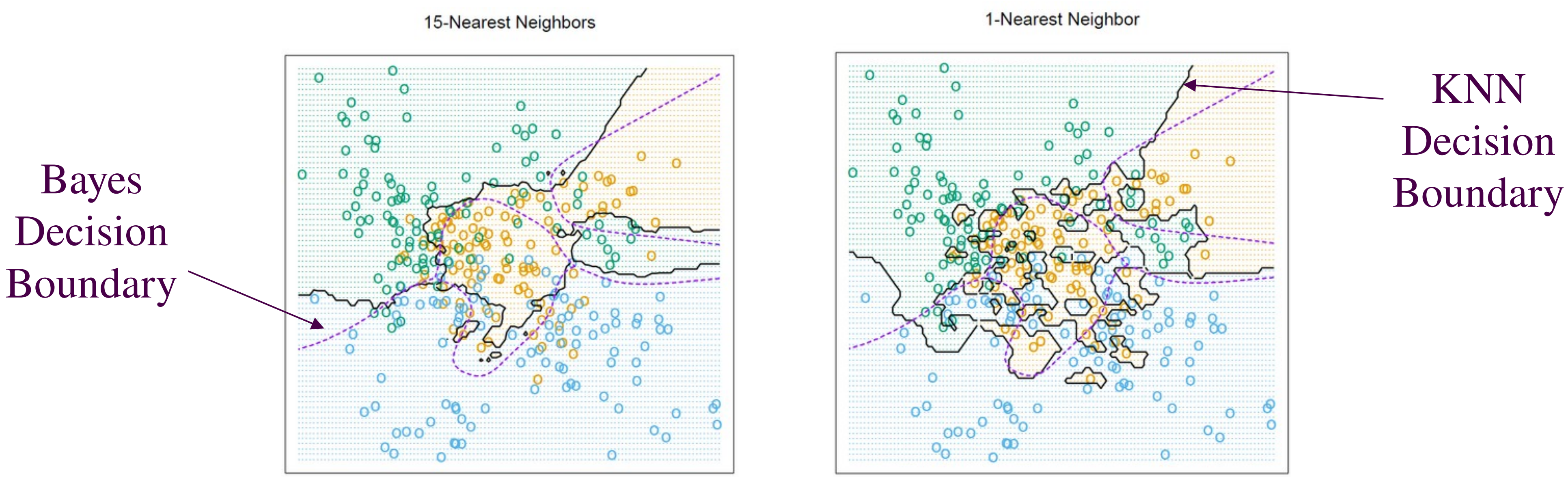
- By looking at only the closest sample, overfitting the data can be a huge problem.
- To prevent overfitting, we can smooth the decision boundary by ***K* nearest neighbors** instead of 1.
- Find the *K* training samples x_r, $r = 1,\ldots,K$ closest in distance to x^*, and then classify using majority vote among the *k* neighbors.
- The amount of computation can be intense when the training data is large since the distance between a new data point and every training point has to be computed and sorted.

K-Nearest Neighbors

- Feature *standardization* is often performed in pre-processing (see our lecture on PCA).
- Because standardization affects the distance, if one wants the features to play a similar role in determining the distance, standardization is recommended.
- However, whether to apply normalization is rather subjective.
- One has to decide on an individual basis for the problem in consideration.

K-Nearest Neighbors

- The only parameter that can adjust the complexity of KNN is the number of neighbors *k*.
- The larger *k* is, the smoother the classification boundary. Or we can think of the complexity of KNN as lower when *k* increases.

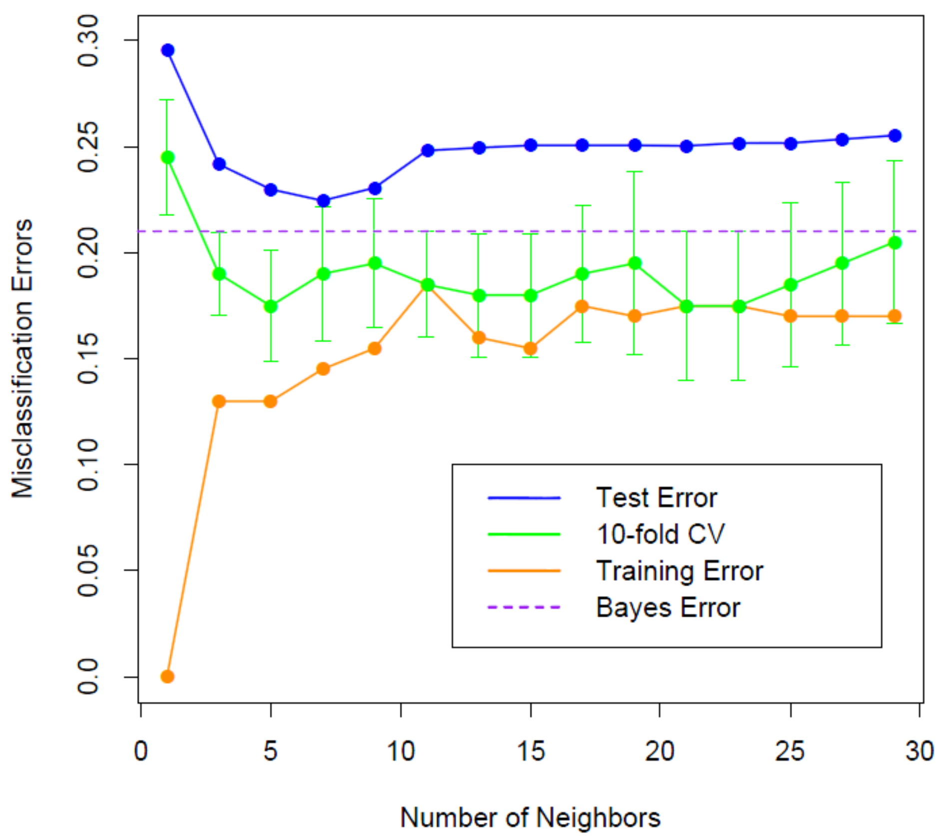


The parameter *k* should be tuned for each problem.

K-Nearest Neighbors

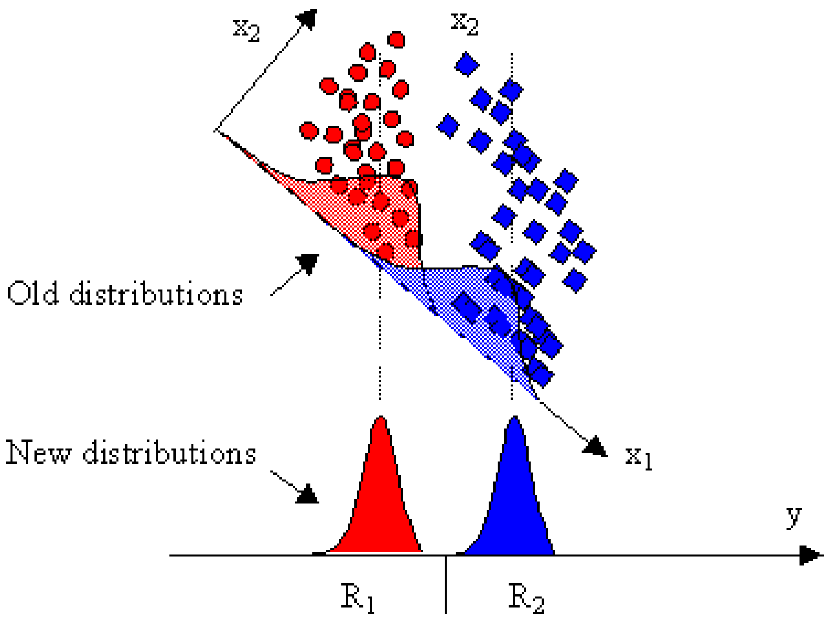
- For another simulated data set, there are two classes. The error rates based on the training data, test data, and *10*-fold cross validation are plotted against *k*, the number of neighbors.

- We can see that the training error rate tends to grow when *k* grows, which is not the case for the error rate based on a separate test data set or cross-validation.



Fisher linear discriminants
LDA, Linear Discriminat Analysis

Projection to one dimension, than discrimination



Equivalent to linear separation

We choose a projection vector in such a way, that the separation is maximized.

Assumptions for new basis:

- Maximize distance between projected class means
- Minimize projected class variance

Method introduced by Fisher in 1936.
Optimal separation for Gaussian distributions.

Fisher Linear Discriminant Analysis

Objective

$$argmax_w J(w) = \frac{w^T S_B w}{w^T S_W w}$$

w - projection of vector
x on 1-dimension

$$m_i = \frac{1}{n_i} \sum_{x \in C_i} x$$

$$S_B = (m_2 - m_1)(m_2 - m_1)^T$$

Variance Between classes

$$S_W = \sum_j^2 \sum_{x \in C_j} (x - m_j)(x - m_j)^T$$

Variance Within class

Algorithm

1. Compute class means
2. Compute $w = S_W^{-1}(m_2 - m_1)$
3. Project data $y = w^T x$

Fisher Linear Discriminant Analysis

A fixed linear combination of the **x**'s takes the values $y_{11}, y_{12}, \ldots, y_{1n_1}$ for the observations from the first population and the values $y_{21}, y_{22}, \ldots, y_{2n_2}$ for the observations from the second population. The separation of these two sets of univariate y's is assessed in terms of the difference between $\bar{y}_1$ and $\bar{y}_2$. expressed in standard deviation units. That is,

$$\text{separation} = \frac{|\bar{y}_1 - \bar{y}_2|}{s_y}, \quad \text{where } s_y^2 = \frac{\sum_{j=1}^{n_1}(y_{1j} - \bar{y}_1)^2 + \sum_{j=1}^{n_2}(y_{2j} - \bar{y}_2)^2}{n_1 + n_2 - 2}$$

is the pooled estimate of the variance. The objective is to select the linear combination of the **x** to achieve maximum separation of the sample means $\bar{y}_1$ and $\bar{y}_2$.

Fisher’s linear discriminant (derivation)

Find the best direction **w** for accurate classification.

A measure of the separation between the projected points is the difference of the sample means.

If m_i is the d-dimensional sample mean from D_i given by:

$$\mathbf{m}_i = \frac{1}{n_i} \sum_{\mathbf{x} \in \mathcal{D}_i} \mathbf{x},$$

The sample mean from the projected points Y_i given by:

$$\tilde{m}_i = \frac{1}{n_i} \sum_{y \in \mathcal{Y}_i} y$$

$$= \frac{1}{n_i} \sum_{\mathbf{x} \in \mathcal{D}_i} \mathbf{w}^t \mathbf{x} = \mathbf{w}^t \mathbf{m}_i$$

The difference of the projected sample means is:

$$|\tilde{m}_1 - \tilde{m}_2| = |\mathbf{w}^t(\mathbf{m}_1 - \mathbf{m}_2)|$$

Fisher's linear discriminant (derivation)

Define ***scatter*** for the projection:

$$\tilde{s}_i^2 = \sum_{y \in \mathcal{Y}_i} (y - \tilde{m}_i)^2.$$

Choose **w** in order to maximize:

$$J(\mathbf{w}) = \frac{|\tilde{m}_1 - \tilde{m}_2|^2}{\tilde{s}_1^2 + \tilde{s}_2^2}$$

$\tilde{s}_1^2 + \tilde{s}_2^2$ is called the total *within-class scatter*.

Define ***scatter matrices*** S_i (i = 1, 2) and S_W by

$$\mathbf{S}_i = \sum_{\mathbf{x} \in \mathcal{D}_i} (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^t \qquad \mathbf{S}_W = \mathbf{S}_1 + \mathbf{S}_2.$$

Fisher's linear discriminant (derivation)

$$\tilde{s}_i^2 = \sum_{\mathbf{x}\in\mathcal{D}_i} (\mathbf{w}^t\mathbf{x} - \mathbf{w}^t\mathbf{m}_i)^2$$

$$= \sum_{\mathbf{x}\in\mathcal{D}_i} \mathbf{w}^t(\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^t\mathbf{w}$$

$$= \mathbf{w}^t\mathbf{S}_i\mathbf{w};$$

We obtain $$\tilde{s}_1^2 + \tilde{s}_2^2 = \underline{\mathbf{w}^t\mathbf{S}_W\mathbf{w}}.$$

39

Fisher's linear discriminant (derivation)

$$|\tilde{m}_1 - \tilde{m}_2| = |\mathbf{w}^t(\mathbf{m}_1 - \mathbf{m}_2)|$$

$$\begin{aligned}(\tilde{m}_1 - \tilde{m}_2)^2 &= (\mathbf{w}^t\mathbf{m}_1 - \mathbf{w}^t\mathbf{m}_2)^2 \\ &= \mathbf{w}^t(\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^t\mathbf{w} \\ &= \underline{\mathbf{w}^t\mathbf{S}_B\mathbf{w}},\end{aligned}$$

where $$\underline{\mathbf{S}_B = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^t}.$$

In terms of S_B and S_w, J(w) can be written as:

$$J(\mathbf{w}) = \frac{\mathbf{w}^t\mathbf{S}_B\mathbf{w}}{\mathbf{w}^t\mathbf{S}_W\mathbf{w}}.$$

Note that S_B and S_W are symmetric.

40

Fisher's linear discriminant (derivation)

$$J(\mathbf{w}) = \frac{\mathbf{w}^t \mathbf{S}_B \mathbf{w}}{\mathbf{w}^t \mathbf{S}_W \mathbf{w}}.$$

Differentiating with respect to **w**, we find that J**(w)** is maximized when:

$$(\mathbf{w}^{\mathrm{T}} \mathbf{S}_{\mathrm{B}} \mathbf{w}) \mathbf{S}_{\mathrm{W}} \mathbf{w} = (\mathbf{w}^{\mathrm{T}} \mathbf{S}_{\mathrm{W}} \mathbf{w}) \mathbf{S}_{\mathrm{B}} \mathbf{w}$$

- $\mathbf{S}_B$ is always in the direction of $\mathbf{m}_1$-$\mathbf{m}_2$
- We can drop the scalar factors ($\mathbf{w}^T\mathbf{S}_B\mathbf{w}$) and ($\mathbf{w}^T S_{\mathbf{w}}\mathbf{w}$) since we are only interested in the direction of **w**

$$S_W w \propto S_B w$$

$$w \propto S_W^{-1} (m_2 - m_1)$$

Maximum separation - max J(w):

$$D^2 = (m_2 - m_1)^T S_W^{-1} (m_2 - m_1)$$

See minimization lemma – next slide

Minimization lemma

Maximization Lemma. Let $\underset{(p\times p)}{\mathbf{B}}$ be positive definite and $\underset{(p\times 1)}{\mathbf{d}}$ be a given vector. Then, for an arbitrary nonzero vector $\underset{(p\times 1)}{\mathbf{x}}$,

$$\max_{\mathbf{x}\neq\mathbf{0}} \frac{(\mathbf{x}'\mathbf{d})^2}{\mathbf{x}'\mathbf{B}\mathbf{x}} = \mathbf{d}'\mathbf{B}^{-1}\mathbf{d} \qquad (2\text{-}50)$$

with the maximum attained when $\underset{(p\times 1)}{\mathbf{x}} = c\underset{(p\times p)}{\mathbf{B}^{-1}}\underset{(p\times 1)}{\mathbf{d}}$ for any constant $c \neq 0$.

Proof. By the extended Cauchy-Schwarz inequality, $(\mathbf{x}'\mathbf{d})^2 \leq (\mathbf{x}'\mathbf{B}\mathbf{x})(\mathbf{d}'\mathbf{B}^{-1}\mathbf{d})$. Because $\mathbf{x} \neq \mathbf{0}$ and $\mathbf{B}$ is positive definite, $\mathbf{x}'\mathbf{B}\mathbf{x} > 0$. Dividing both sides of the inequality by the positive scalar $\mathbf{x}'\mathbf{B}\mathbf{x}$ yields the upper bound

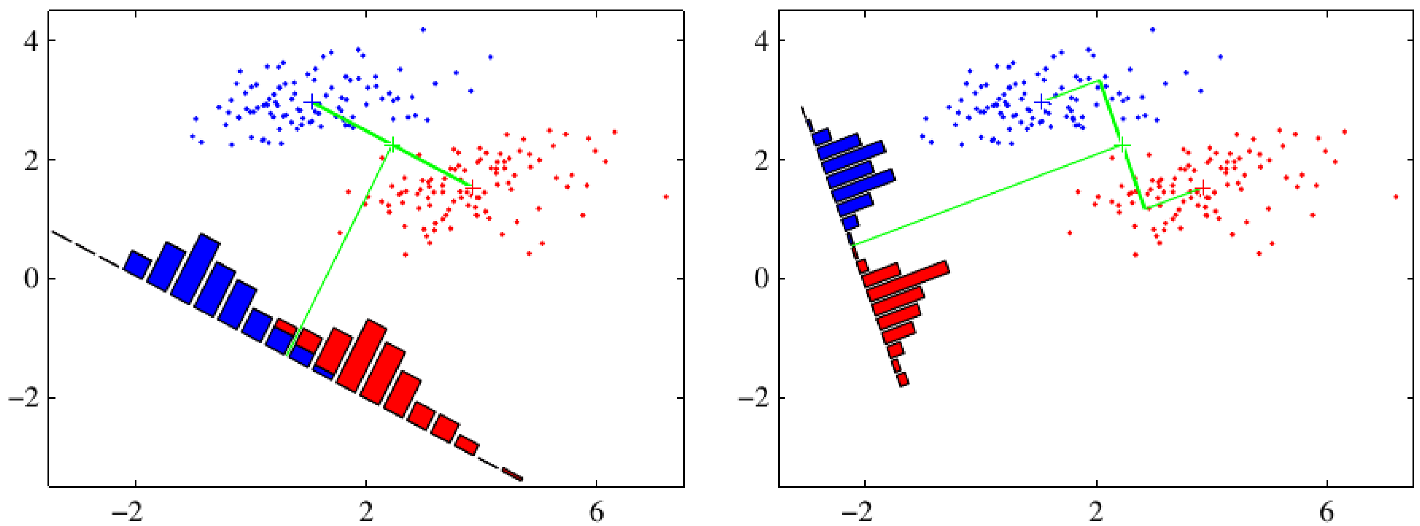
$$\frac{(\mathbf{x}'\mathbf{d})^2}{\mathbf{x}'\mathbf{B}\mathbf{x}} \leq \mathbf{d}'\mathbf{B}^{-1}\mathbf{d}$$

Taking the maximum over $\mathbf{x}$ gives Equation (2-50) because the bound is attained for $\mathbf{x} = c\mathbf{B}^{-1}\mathbf{d}$. ∎

A final maximization result will provide us with an interpretation of eigenvalues.

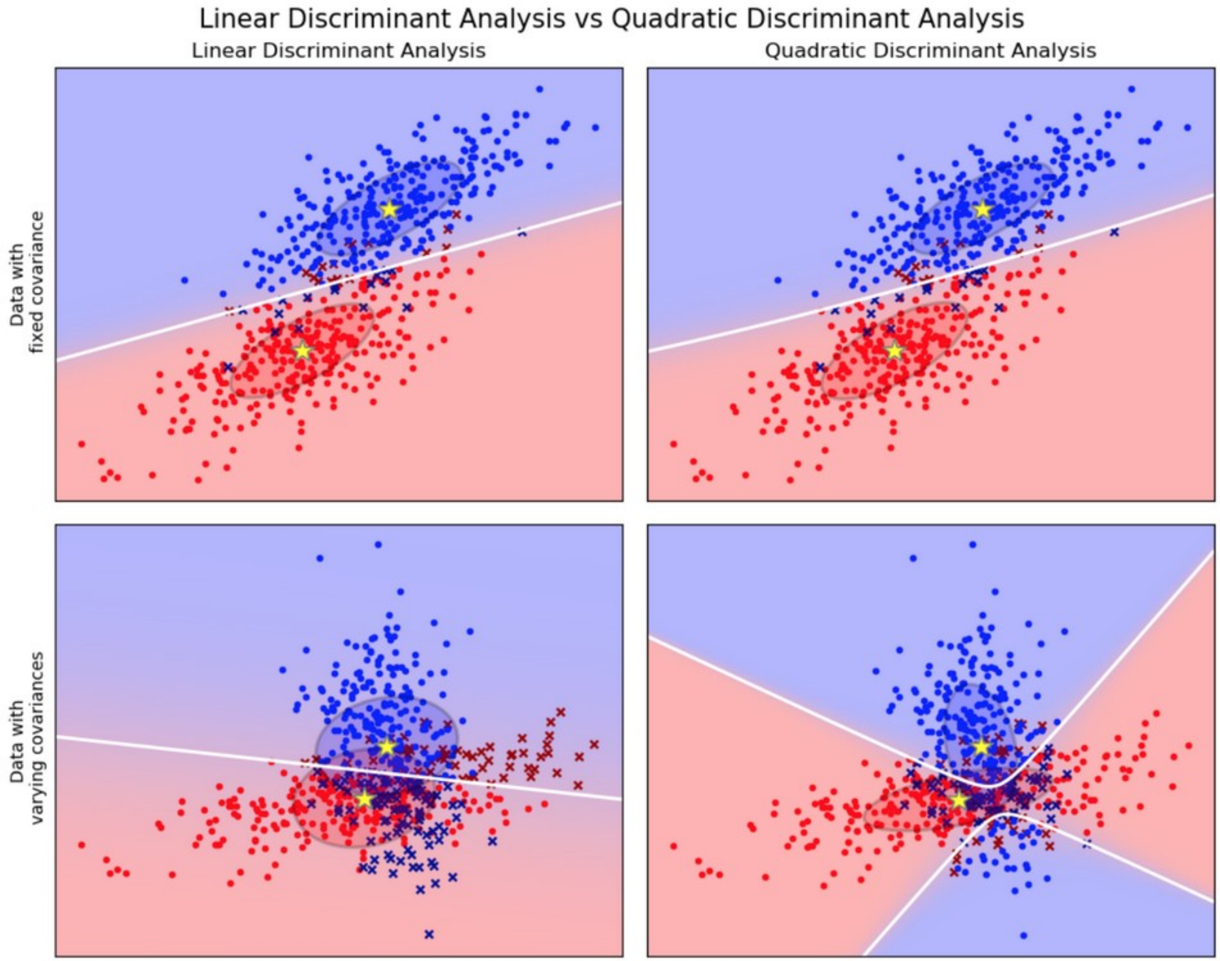
Fisher's linear discriminant

- This result is known as Fisher's linear discriminant
- Strictly it is a specific choice of direction for projection of the data down to one dimension
- The projected data can be used to construct a discriminant by choosing a threshold y_0 so that we classify a new point as belonging to C_1 if $y(x)$ y_0 and classify it as belonging to C_2 otherwise.
-



Quadratic Discriminant Analysis

- In the case of LDA, the Gaussians for each class are assumed to share the same covariance matrix.
- In the case of QDA, there are no assumptions on the covariance matrices of the Gaussians, leading to quadratic decision surfaces.



Python examples

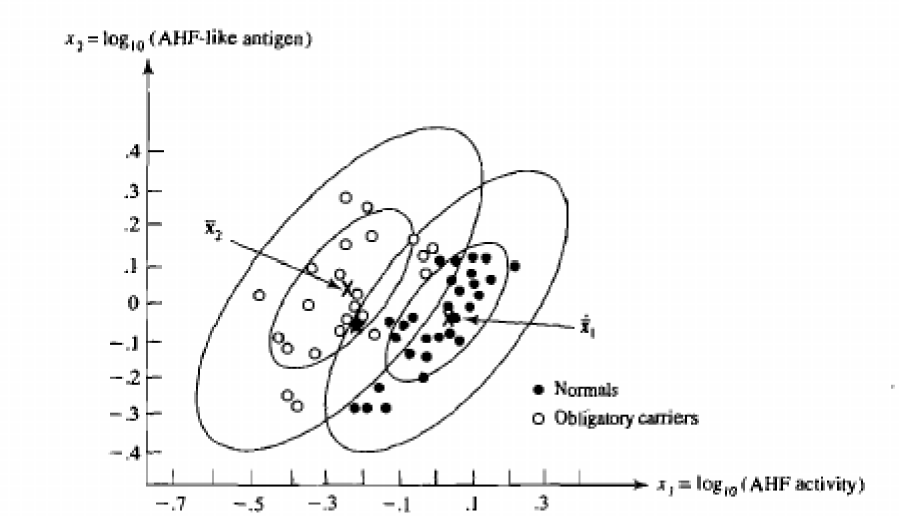
https://github.com/marcinwolter/ANOVA_2019/blob/master/plot_face_recognition.ipynb

https://github.com/marcinwolter/ANOVA_2019/blob/master/simple_classifier_comparison.ipynb

Summary

- We have learned about simple classifiers
- Next lecture – Deep Neural Networks – highly non-linear classifiers

Exercise



Hemophilia studies

How well could we separate these classes?

$$\bar{\mathbf{x}}_1 = \begin{bmatrix} -.0065 \\ -.0390 \end{bmatrix}, \qquad \bar{\mathbf{x}}_2 = \begin{bmatrix} -.2483 \\ .0262 \end{bmatrix} \qquad \mathbf{S}_{\text{pooled}}^{-1} = \begin{bmatrix} 131.158 & -90.423 \\ -90.423 & 108.147 \end{bmatrix}$$

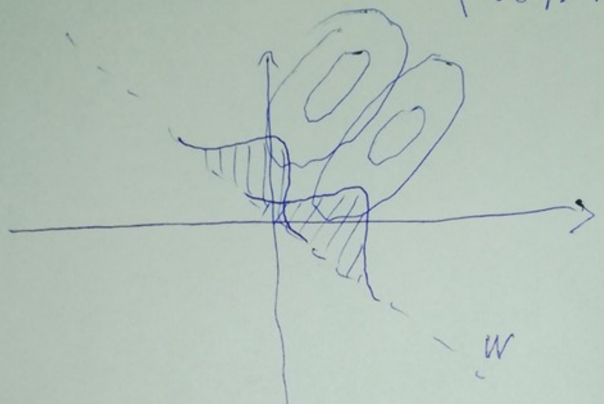
S_W^{-1}

Find the direction of the w vector and the maximal separation.

Kierunek wektora $\vec{w}$:

$$w \propto S_W^{-1}(m_2 - m_1) = \begin{pmatrix} 131,158 & -90,423 \\ -90,423 & 108,147 \end{pmatrix} \begin{pmatrix} -0,2418 \\ 0,0652 \end{pmatrix} =$$

$$= \begin{pmatrix} -37,6096 \\ 28,91 \end{pmatrix}$$



$$D^2 = (m_2 - m_1)^T S_W^{-1} (m_2 - m_1) =$$

$$= (-0,2418 \;;\; 0,0652) \begin{pmatrix} -37,6096 \\ 28,91 \end{pmatrix} = 10,98$$

max. wartość $\dfrac{\text{„wariancja between"}}{\text{„wariancja within"}} = 10,98$