flusser@utia.cas.cz

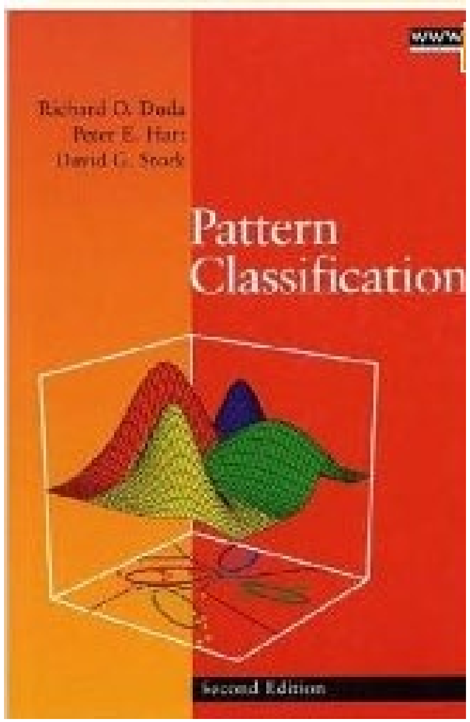
www.utia.cas.cz/people/flusser

Prof. Ing. Jan Flusser, DrSc.

Lectures 2 and 3 – Classifiers

Textbook

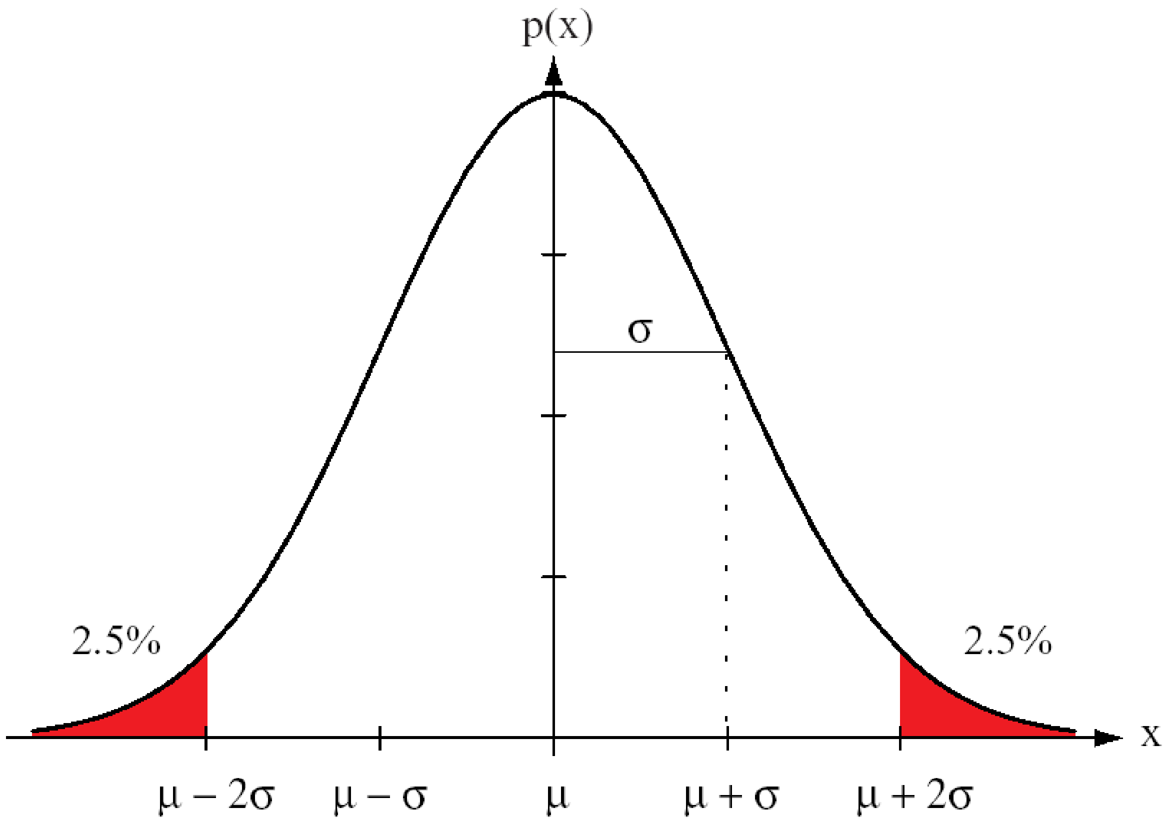
Duda, Hart, Stork: Pattern Classification,
2nd ed., Wiley, 2001



Opakování statistiky

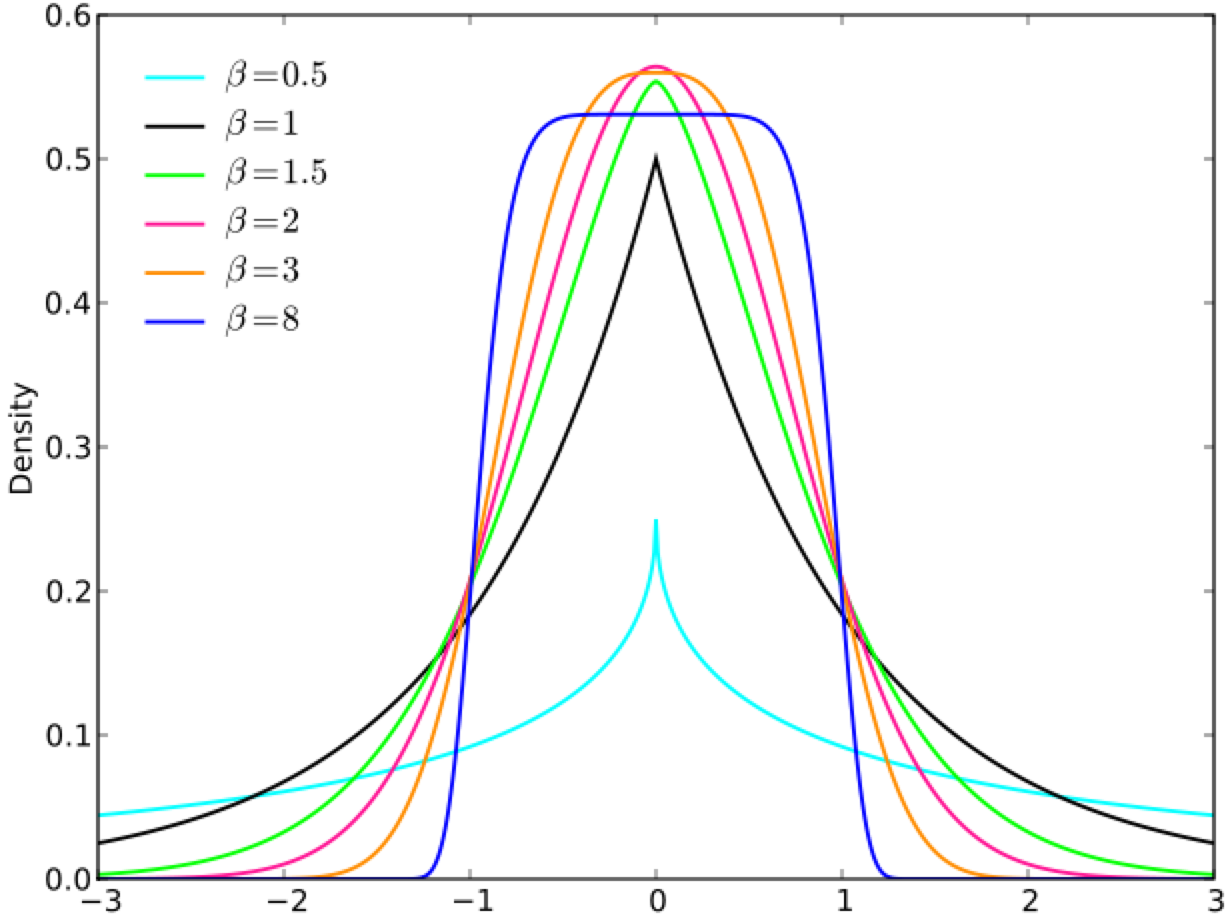
- Pravděpodobnost
- Podmíněná pravděpodobnost
- Náhodná veličina, distribuční fce, hustota
- Střední hodnota
- Rozptyl
- Korelace a kovariance
- Normální rozdělení
- Metoda nejmenších čtverců
- Konvoluce

Normální rozdělení

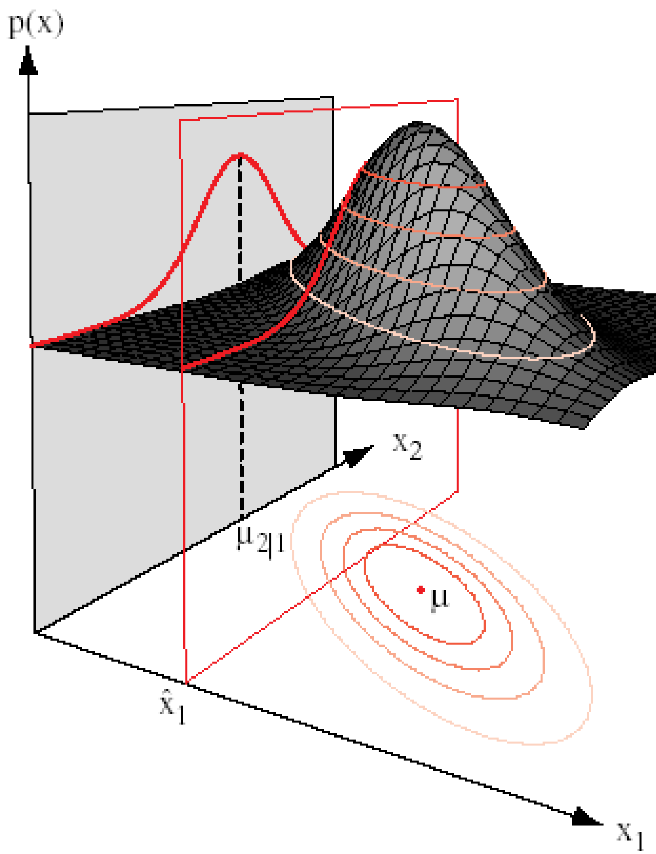


$$p(x) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right]$$

Zobecněné normální rozdělení



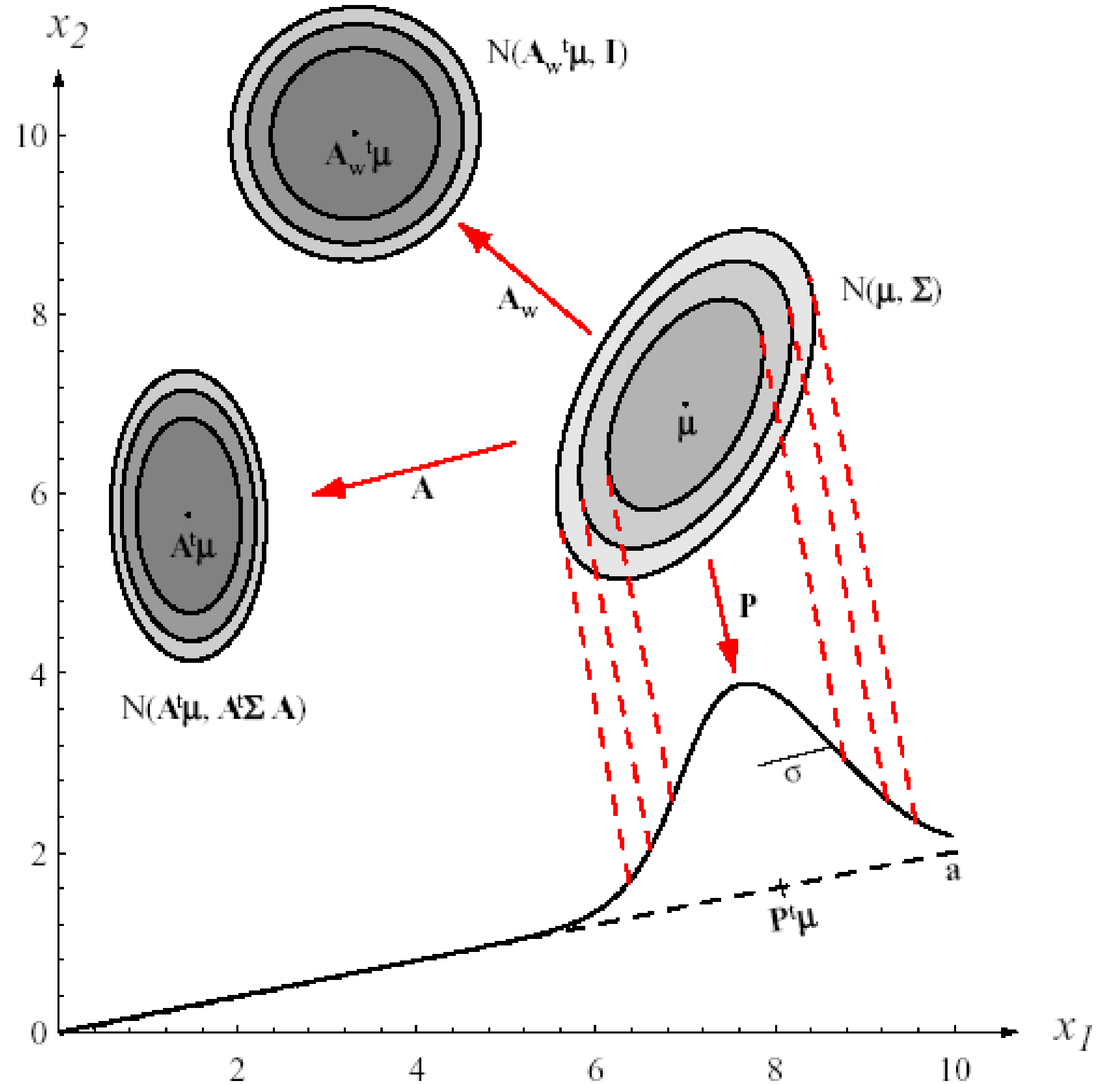
Vícerozměrné normální rozdělení



$$\hat{\mu} = \frac{1}{n}\sum_{k=1}^{n} \mathbf{x}_k$$

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{n}\sum_{k=1}^{n} (\mathbf{x}_k - \hat{\mu})(\mathbf{x}_k - \hat{\mu})^t.$$

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^t\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right]$$



x2
10
8
6
4
2
0
N(Awtμ, I)
Awtμ
N(μ, Σ)
Aw
A
μ
Atμ
N(Atμ, AtΣ A)
P
σ
a
Ptμ
x1
2
4
6
8
10

Normální rozdělení – je nebo není všude?

- Zkušenost ze života
- Centrální limitní věta
- Rozdělení chyb měření

Ovšem není úplně všude (jiná rozdělení, směsi, omezení hodnot, ...)

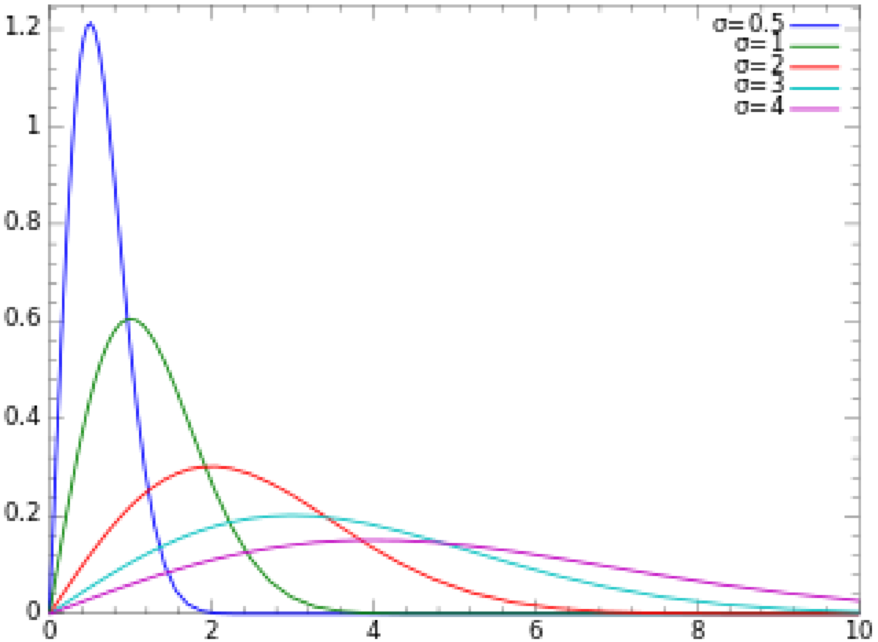
"Jsou tato data z normálního rozdělení?"

Gaussova funkce

- Normální rozdělení
- Vedení tepla, difuze
- Vlastní funkce FT, minimalizuje neurčitost
- Derivace
- Uzavřenost k součinu a konvoluci
- Rozmazání turbulencí atmosféry
- Maximalizuje entropii

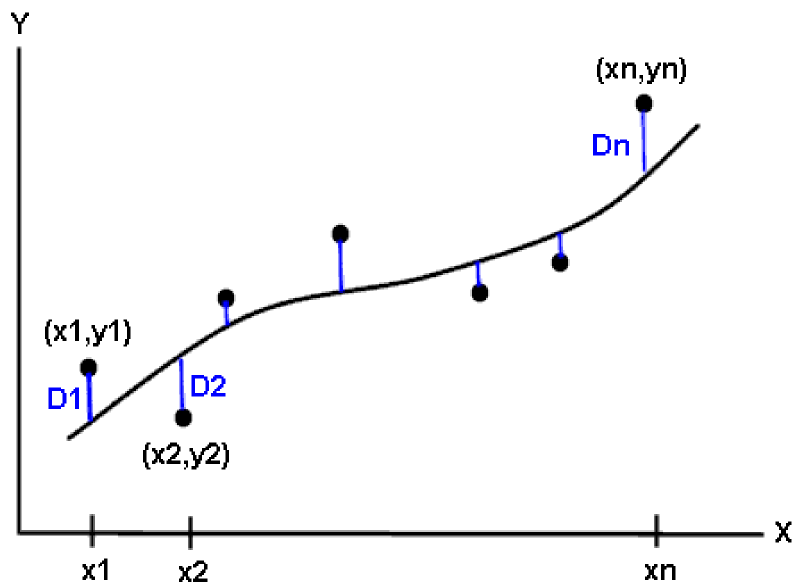
Příklad – střelba do terče

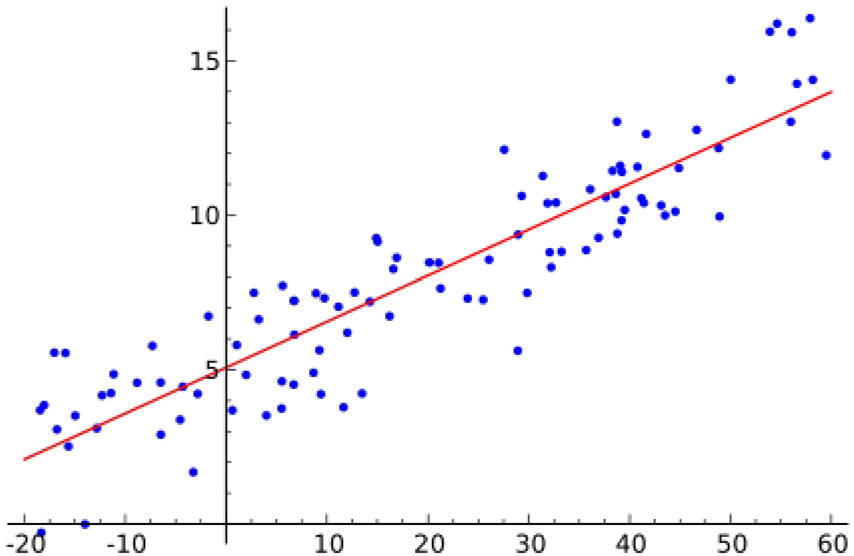
Rayleigh distribution



Metoda nejmenších čtverců

- Aproximace dat (regrese)
- Řešení přezadaných soustav rovnic

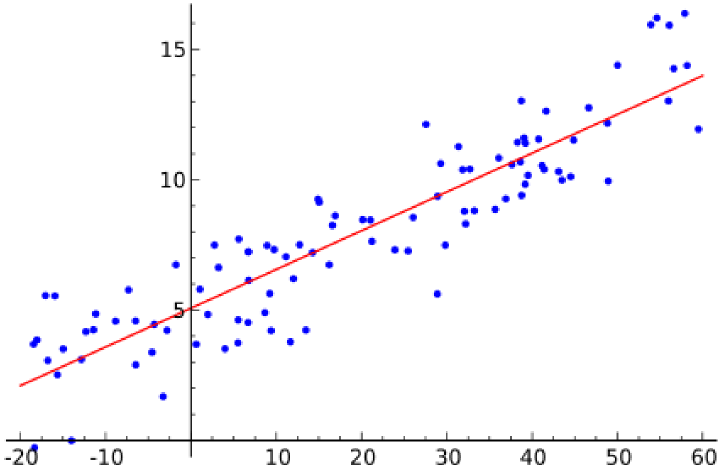




Metoda nejmenších čtverců - základní

- Minimalizace součtu kvadrátů odchylek

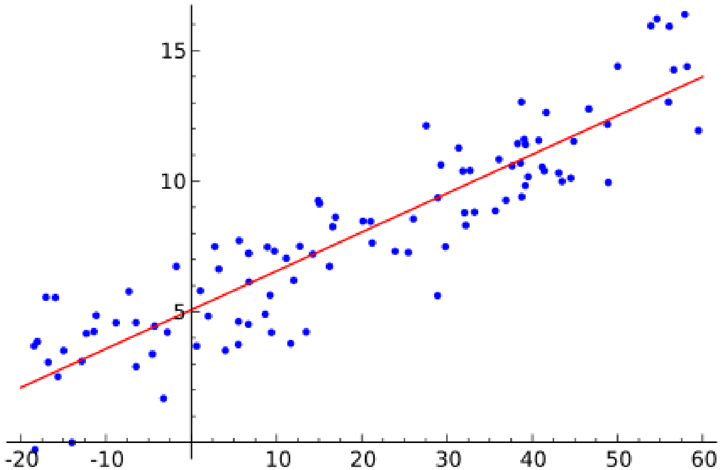
$$E(a,b) = \sum_{n=1}^{N} (y_n - (ax_n + b))^2$$



$$\frac{\partial E}{\partial a} = 0, \quad \frac{\partial E}{\partial b} = 0.$$

Metoda nejmenších čtverců

$$\begin{pmatrix} \sum_{n=1}^{N} x_n^2 & \sum_{n=1}^{N} x_n \\ \sum_{n=1}^{N} x_n & \sum_{n=1}^{N} 1 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \sum_{n=1}^{N} x_n y_n \\ \sum_{n=1}^{N} y_n \end{pmatrix}$$

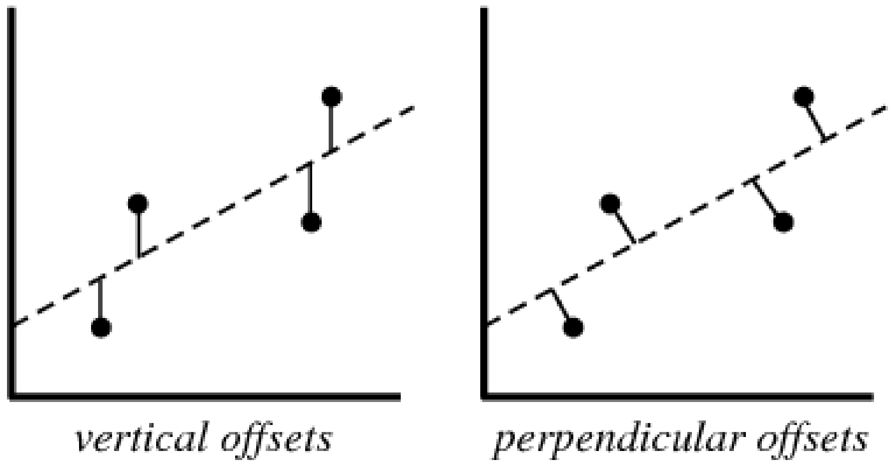


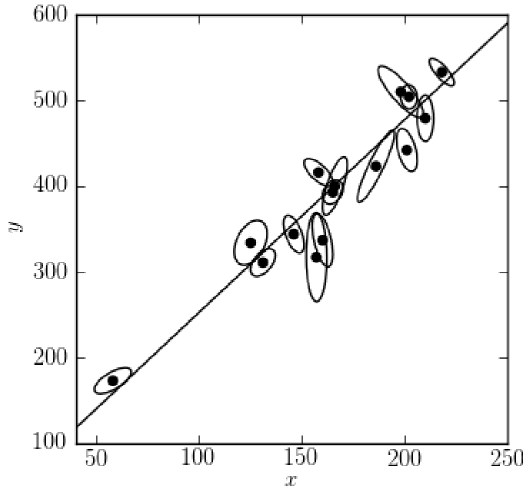
Metoda nejmenších čtverců

- Aproximace polynomem vždy vede na lineární úlohu
- Nelineární MNČ – numerické řešení soustavy rovnic

Totální metoda nejmenších čtverců

- Obě veličiny měříme s chybou
- Matematicky komplikovanější





Pattern Recognition

- Recognition (classification) = assigning a pattern/object to one of pre-defined classes
- Syntactic (structural) PR - the pattern is described by its structure. Formal language theory (class = language, pattern = word)

Pattern Recognition

- Recognition (classification) = assigning a pattern/object to one of pre-defined classes
- Statistical (feature-based) PR - the pattern is described by features (n-D vector in a metric space)

Pattern Recognition

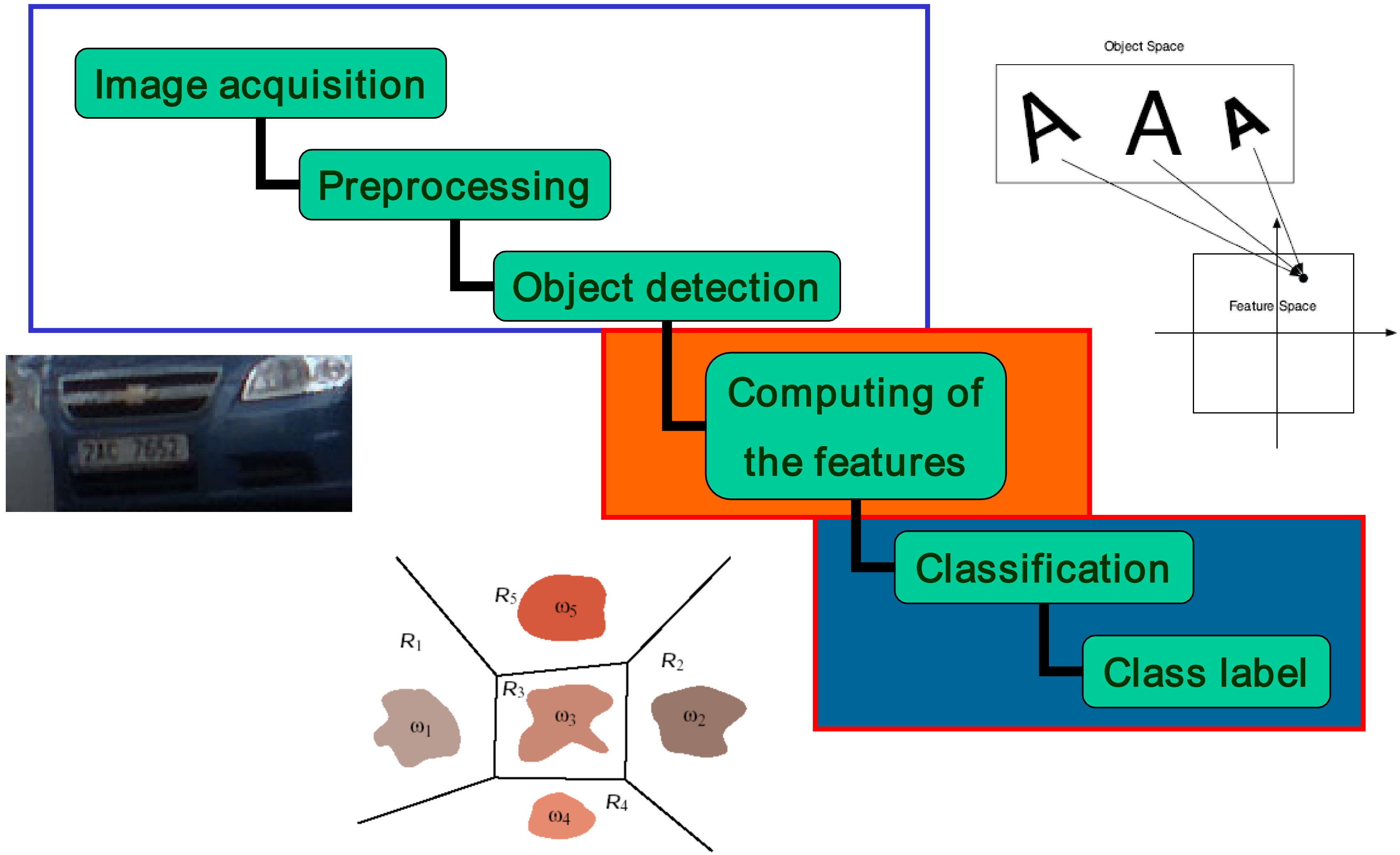
- Supervised PR
  - training set available for each class

- Unsupervised PR (clustering)
  - training set not available, No. of classes may not be known

Supervised Classification

- Classification algorithms work in the feature space and are independent of the data type.
- In ROZ2, we do not review “deep learning” approaches

Object Recognition System

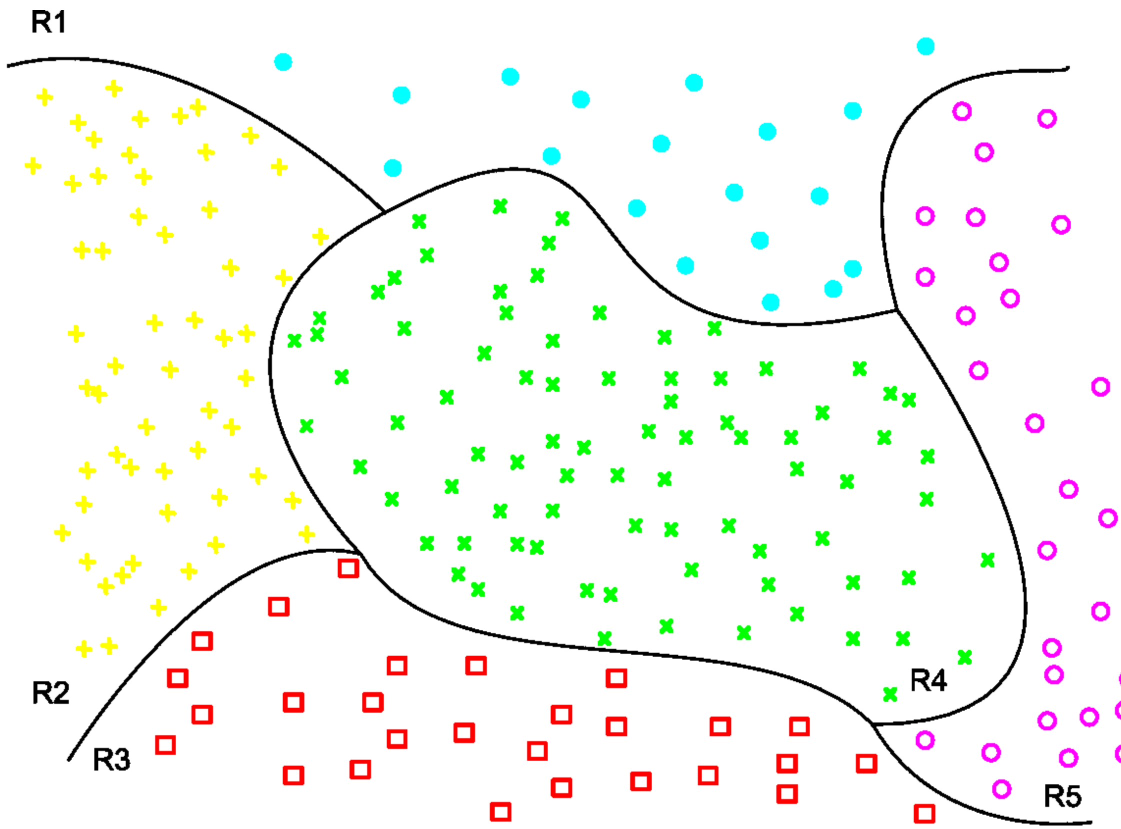


Desirable properties of the training set

- It should contain typical representatives of each class including intra-class variations
- Reliable and large enough
- Should be selected by the domain experts

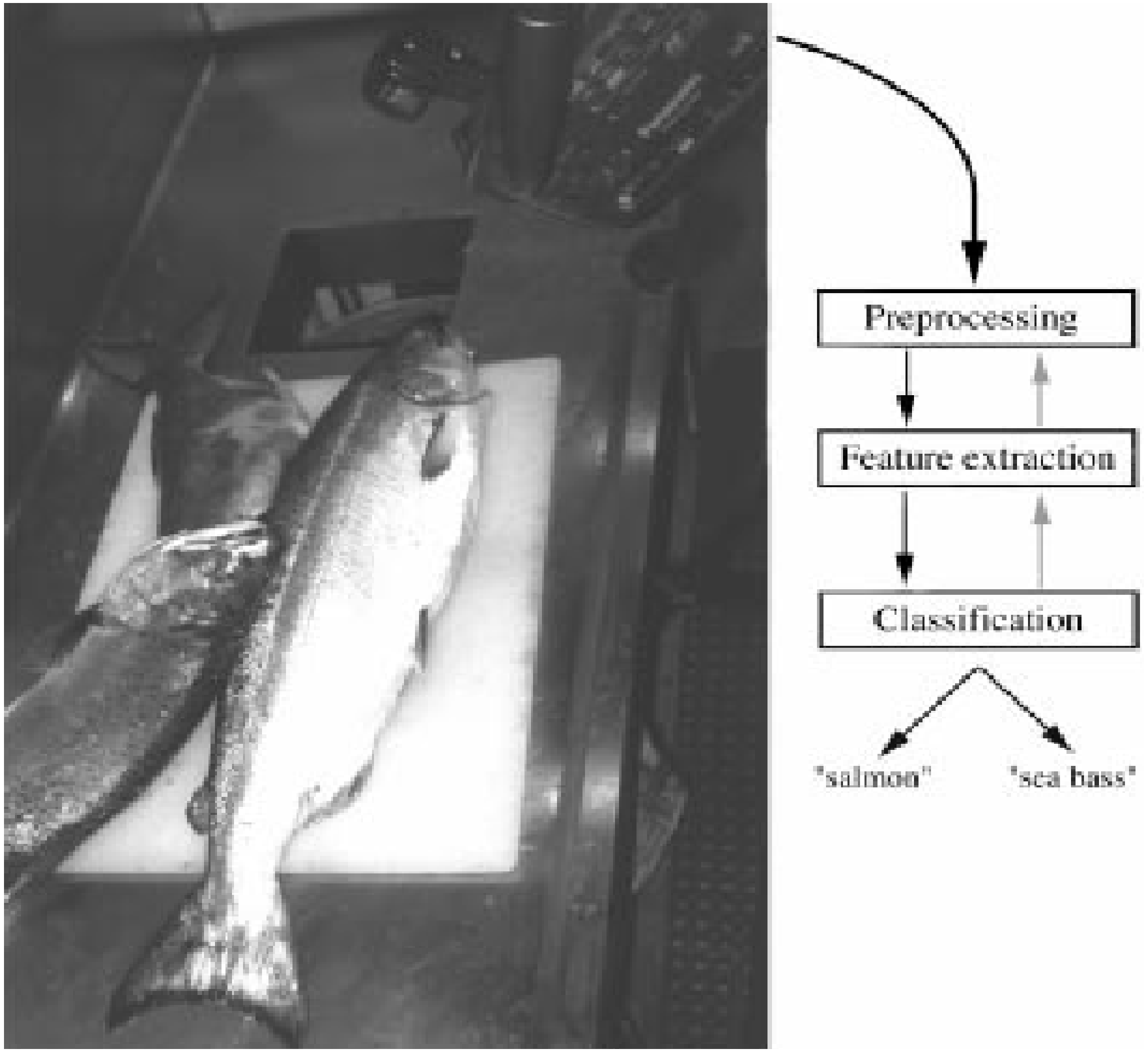
Classification rule setup

Equivalent to a partitioning of the feature space

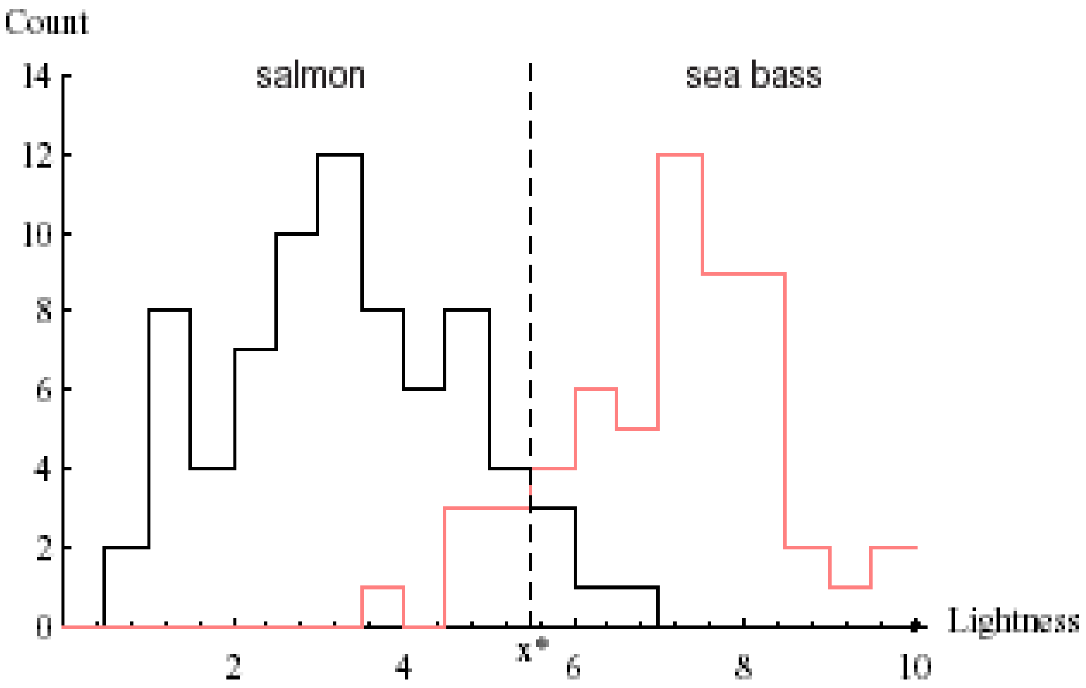


Independent of the particular application

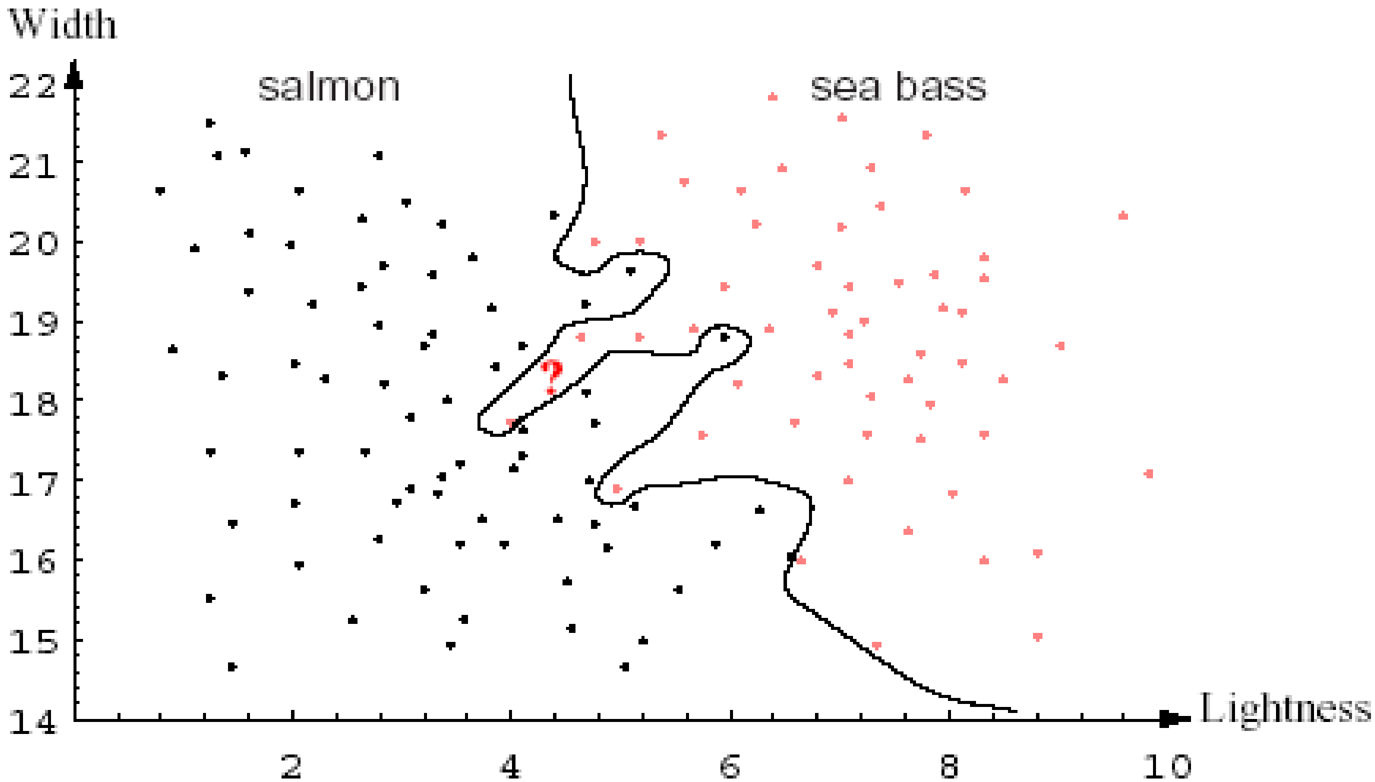
An example – Fish classification



The features: Length, width, brightness



2-D feature space



Empirical observation

- For a given training set, we can have several classifiers (several partitioning of the feature space)

Empirical observation

- For a given training set, we can have several classifiers (several partitioning of the feature space)
- The training samples are not always classified correctly

Empirical observation

- For a given training set, we may have several classifiers (several partitioning of the feature space)
- The training samples are not always classified correctly
- We should avoid overtraining of the classifier

Formal definition of the classifier

- Each class is characterized by its discriminant function *g(x)*
- Classification = maximization of *g(x)*
  Assign *x* to class *i* iff

$$g_i(\mathbf{x}) > g_j(\mathbf{x}) \qquad \forall j \neq i$$

- Discriminant functions define decision boundaries in the feature space

Minimum distance (NN) classifier

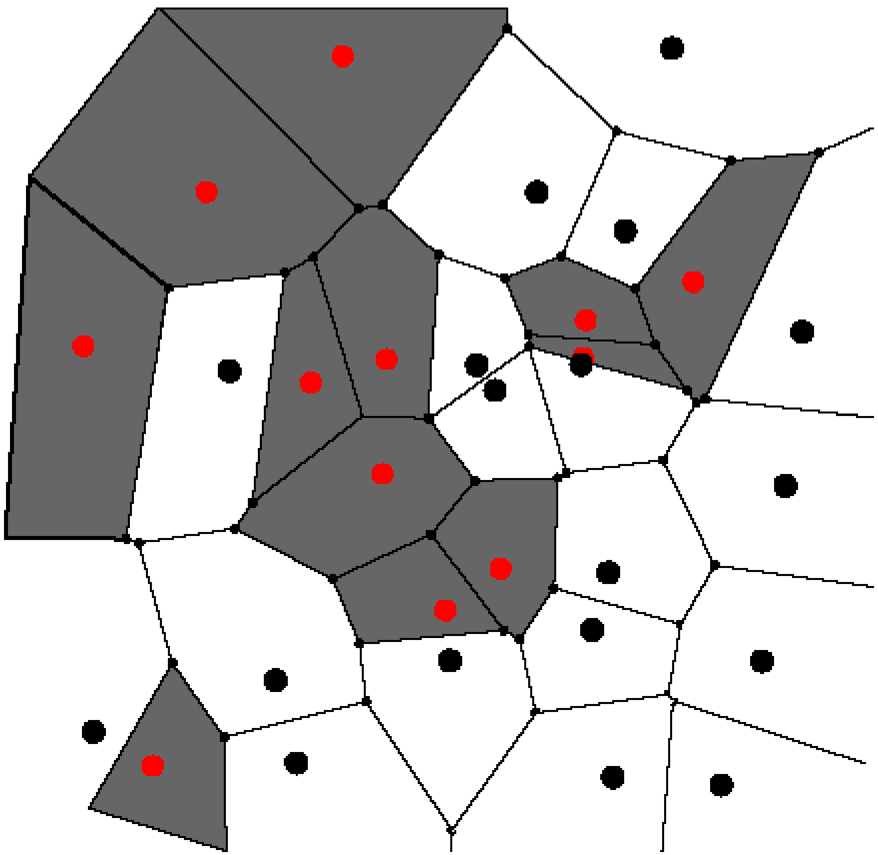
- Discriminant function

$$g_i(\mathbf{x}) = -d(\mathbf{x}, \omega_i)$$

- Various definitions of $d(\mathbf{x}, \omega_i)$

- One-element training set →

Voronoi polygons

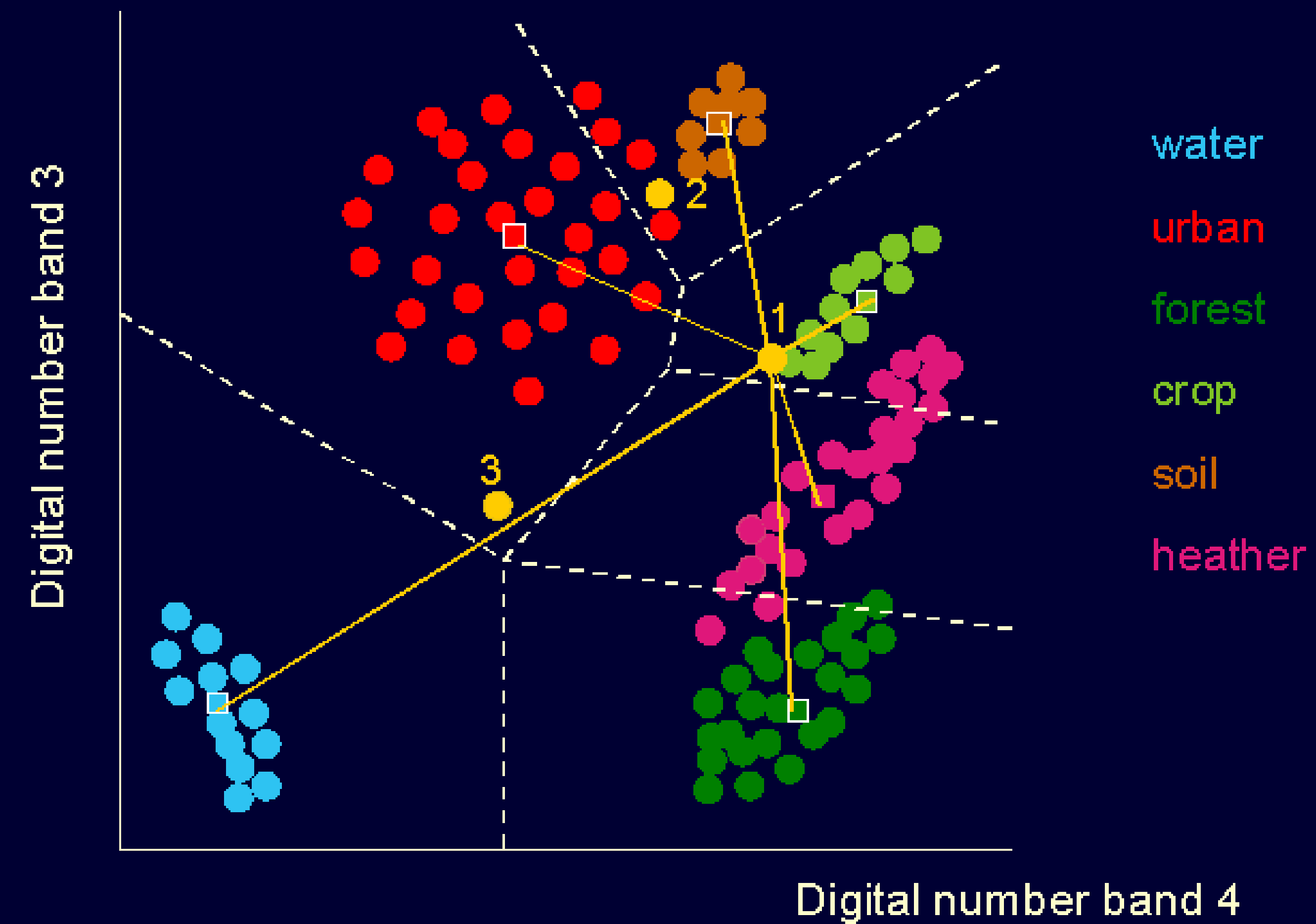


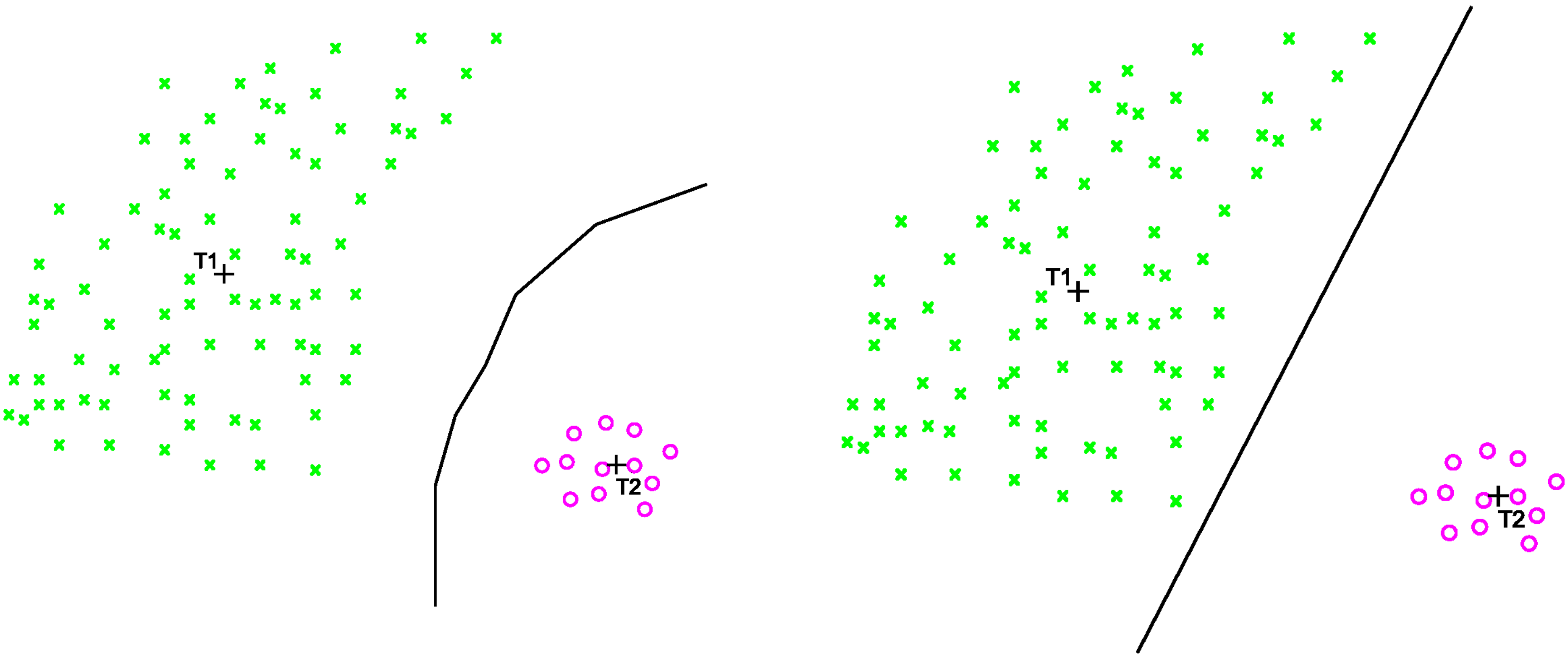
Minimum distance (NN) classifier

- Depending on $d(\mathbf{x}, \omega_i)$, NN classifier may not be linear

- NN classifier is sensitive to outliers → k-NN classifier

Minimum distance to means classification



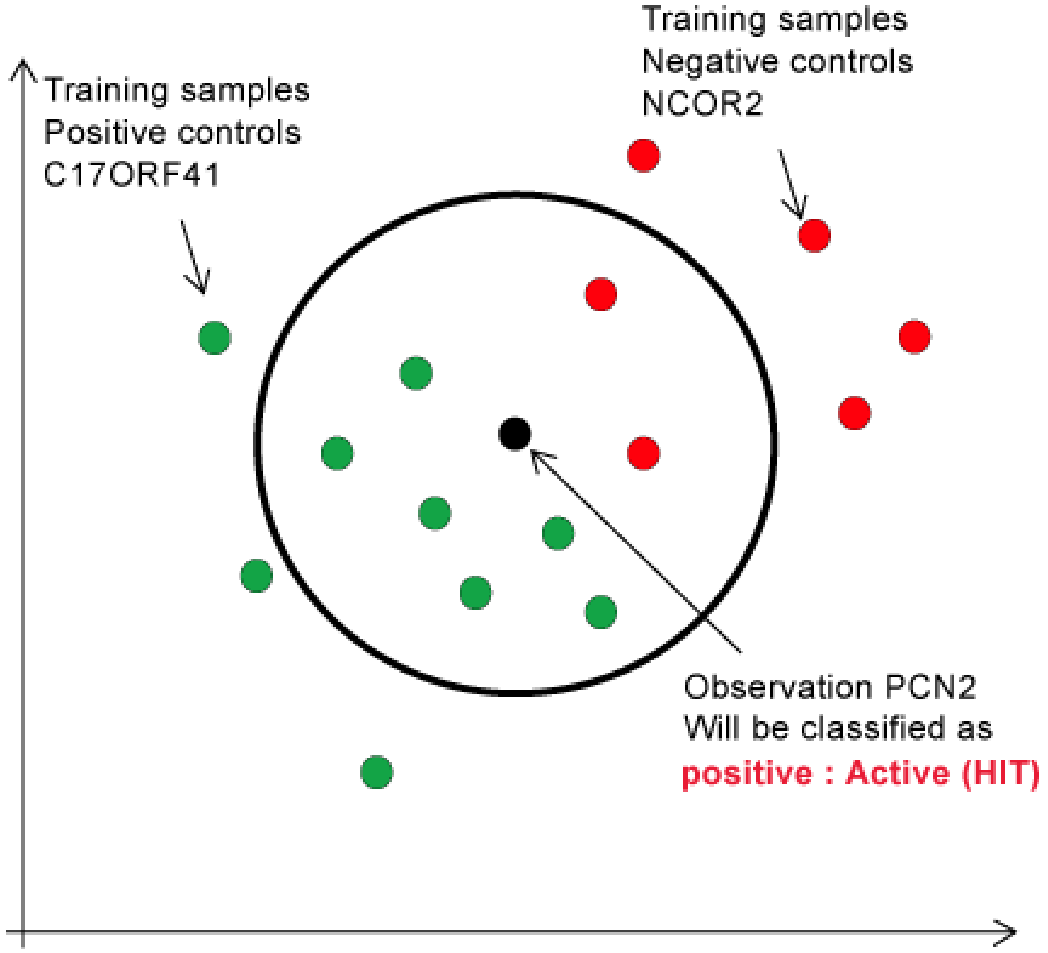

T1
T2
T1
T2

k- NN classifier

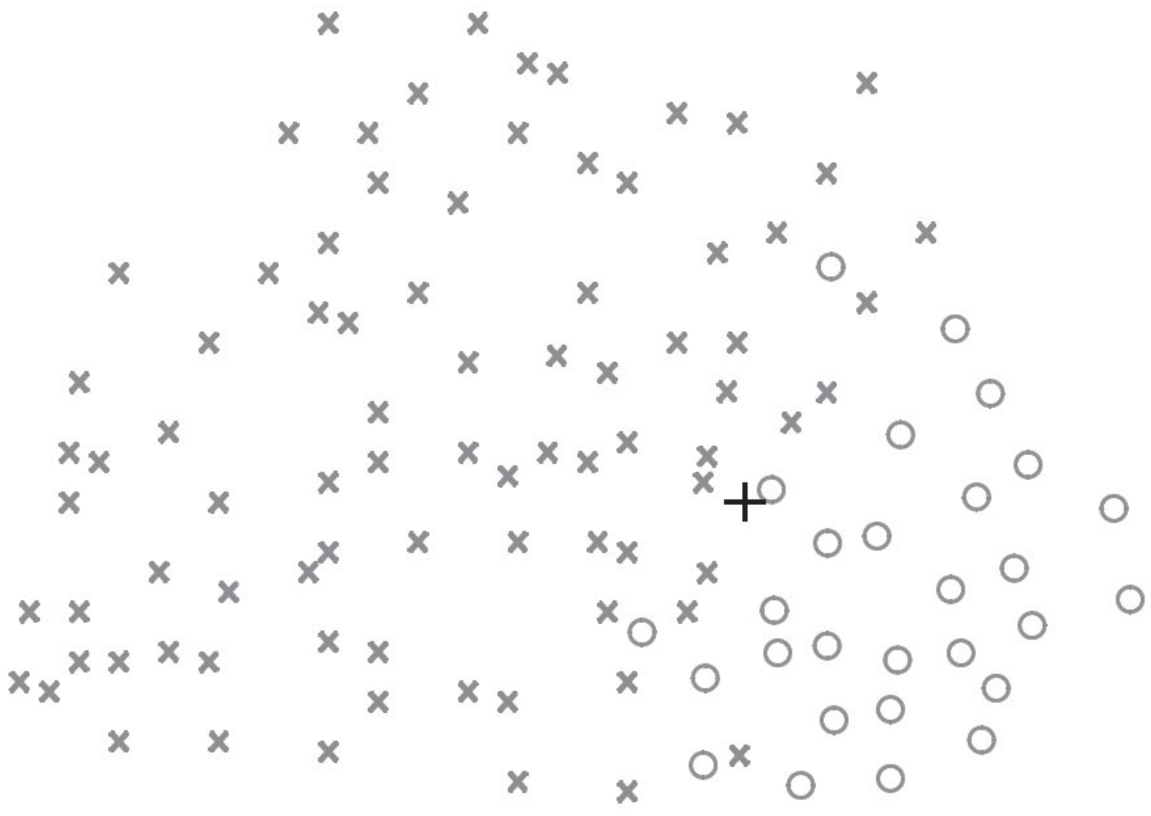
- NN classifier is sensitive to outliers → k-NN classifier

- It finds the nearest training points unless *k* samples belonging to one class has been reached

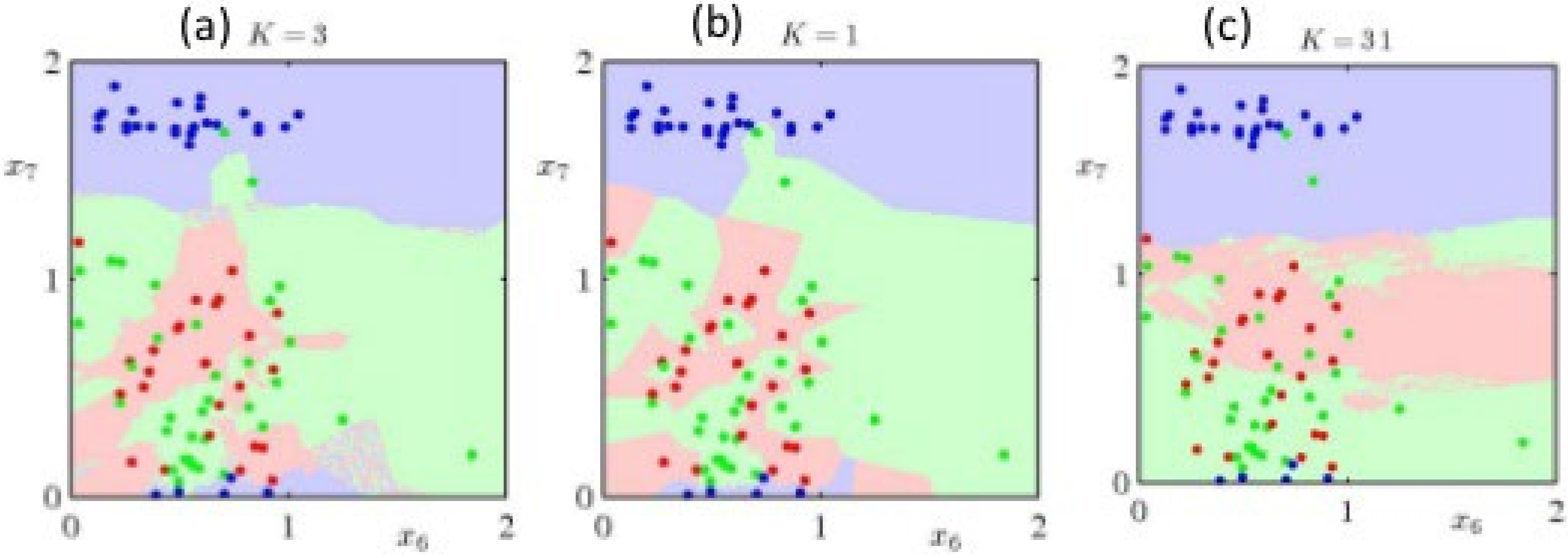
k-NN classifier



k-NN classifier

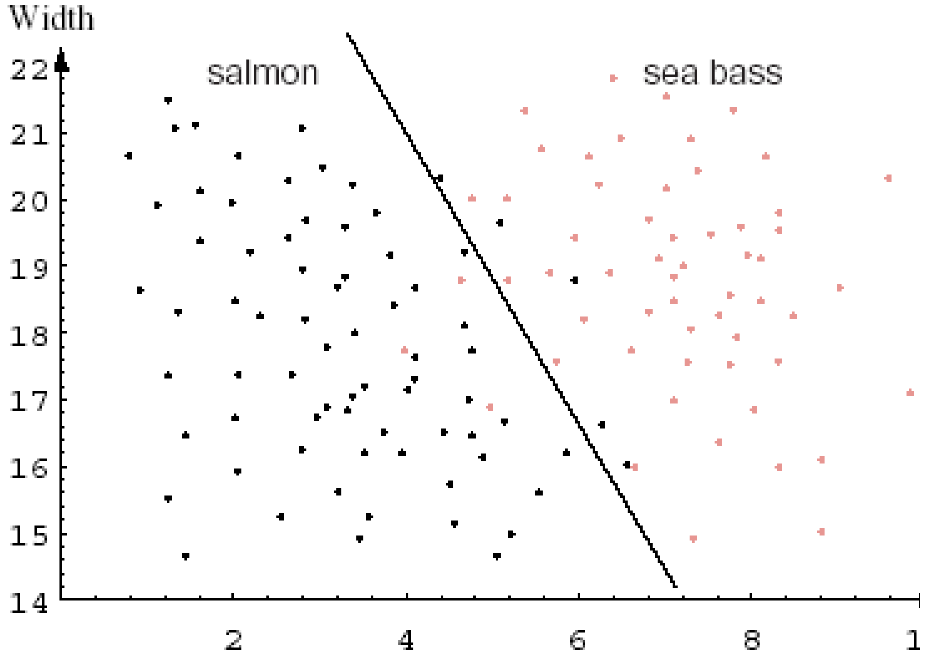


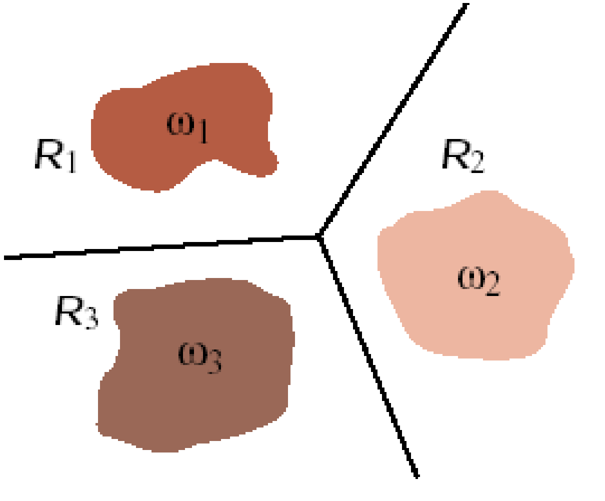
k-NN classifier



Linear classifier

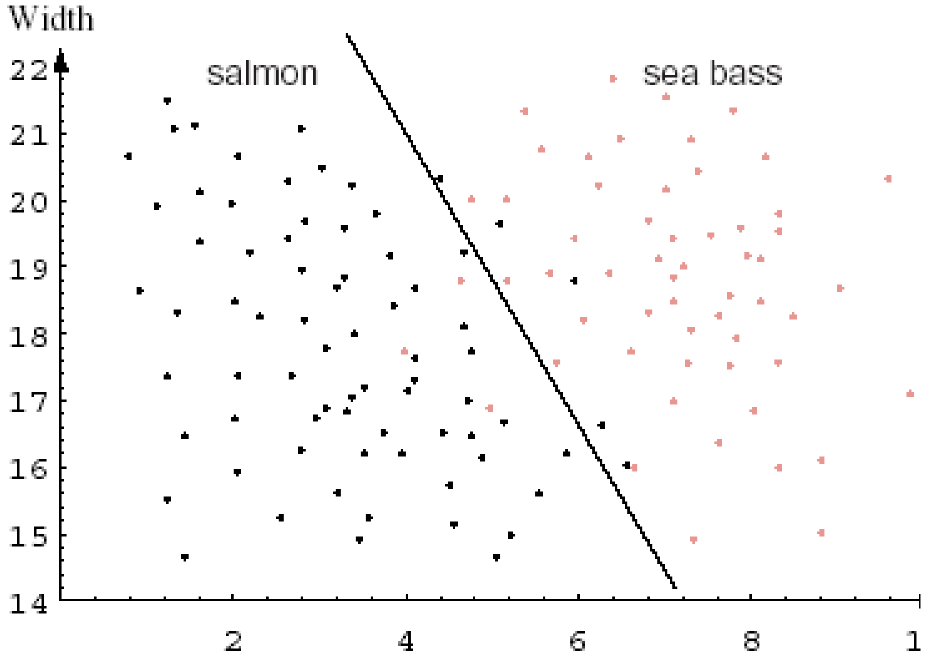
Discriminant functions g(x) are hyperplanes





Simple training algorithms

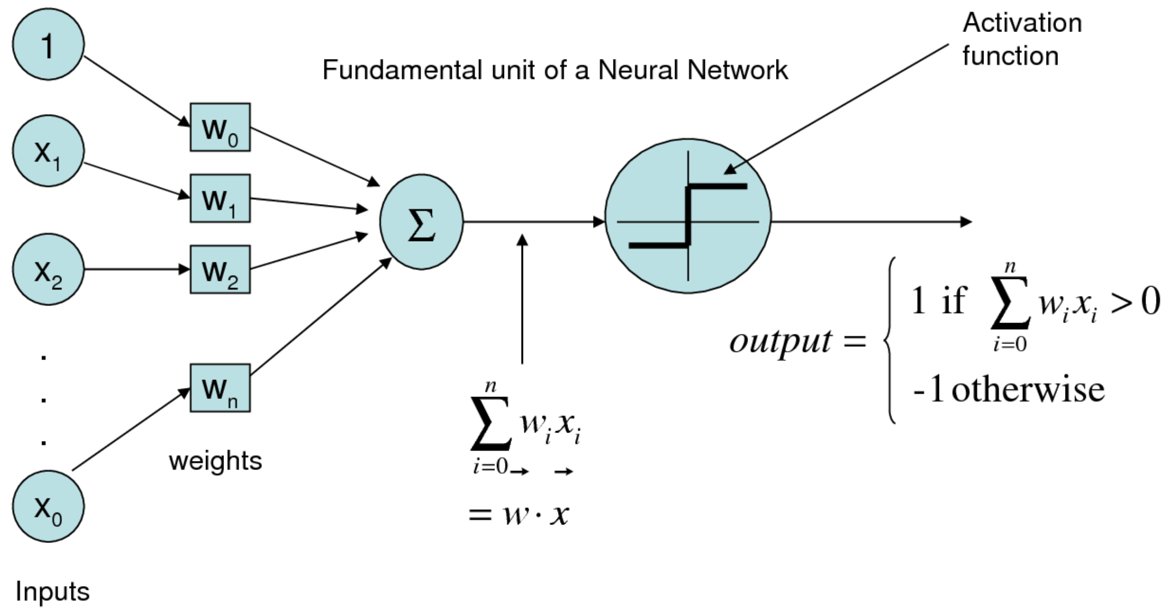
- Many possible hyperplanes
- Perceptron



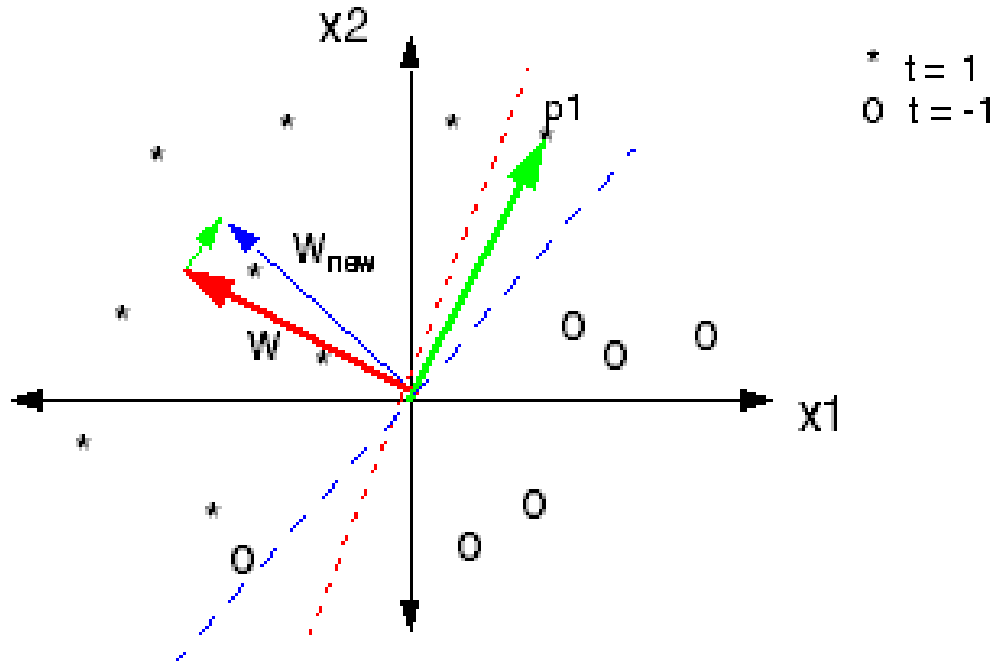
Perceptron

Artificial Neural Networks

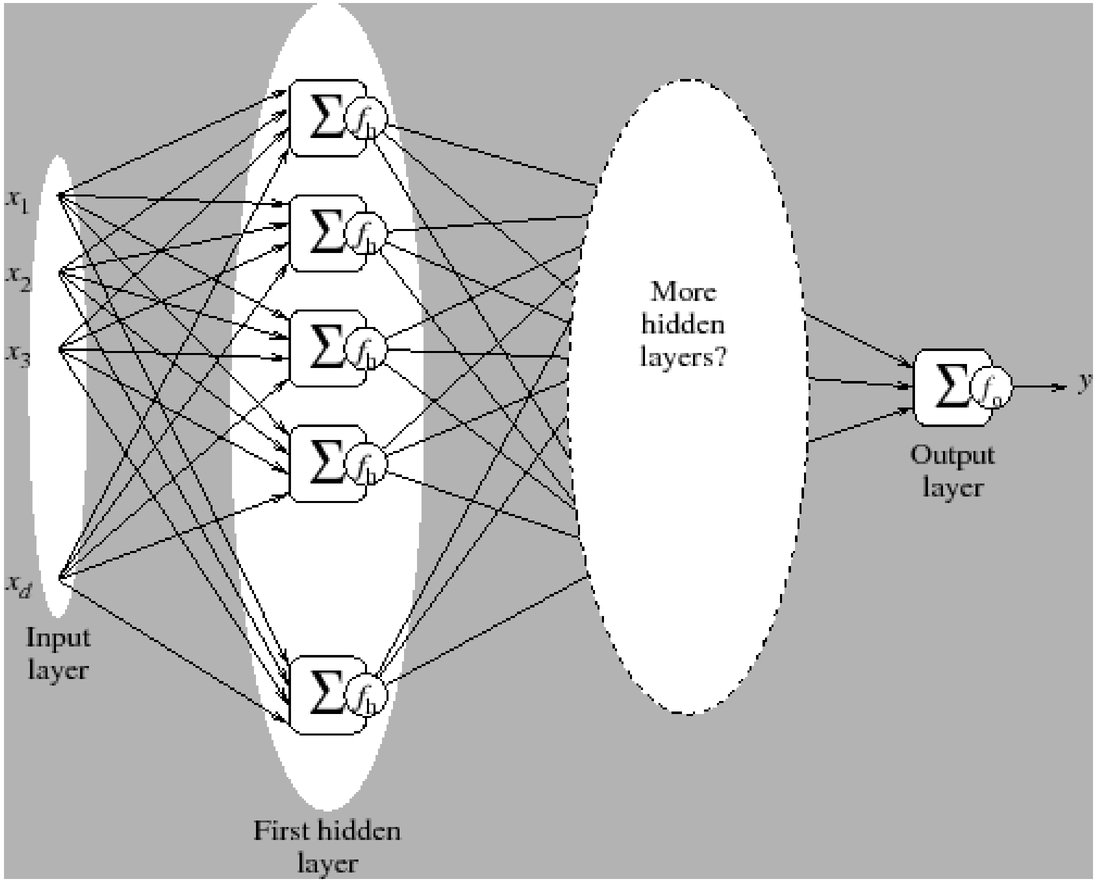
The Perceptron



Perceptron – an iterative training

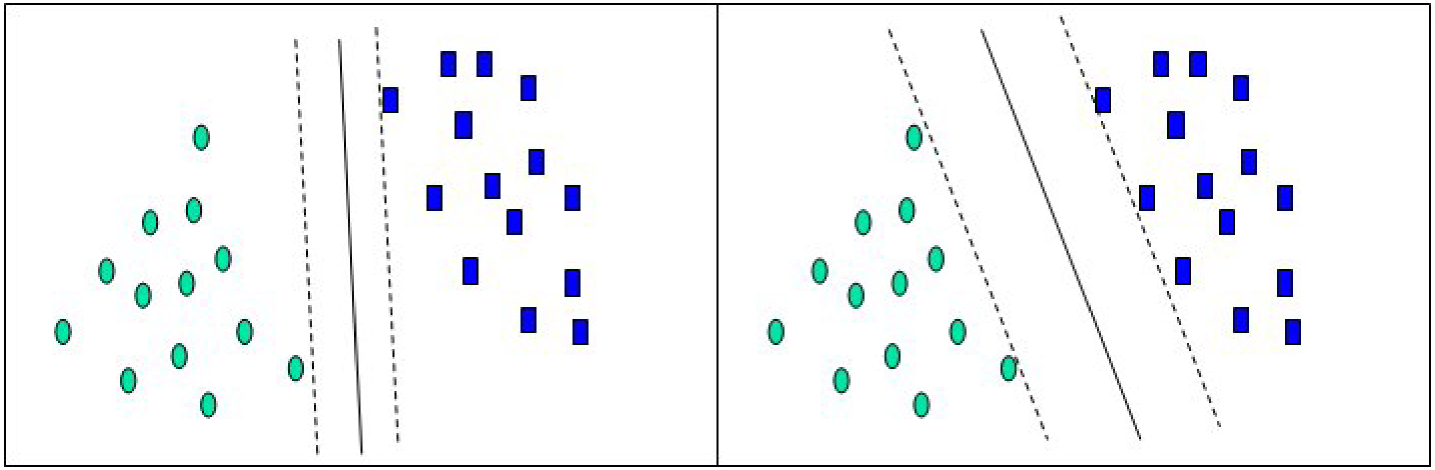


Multilayer perceptron



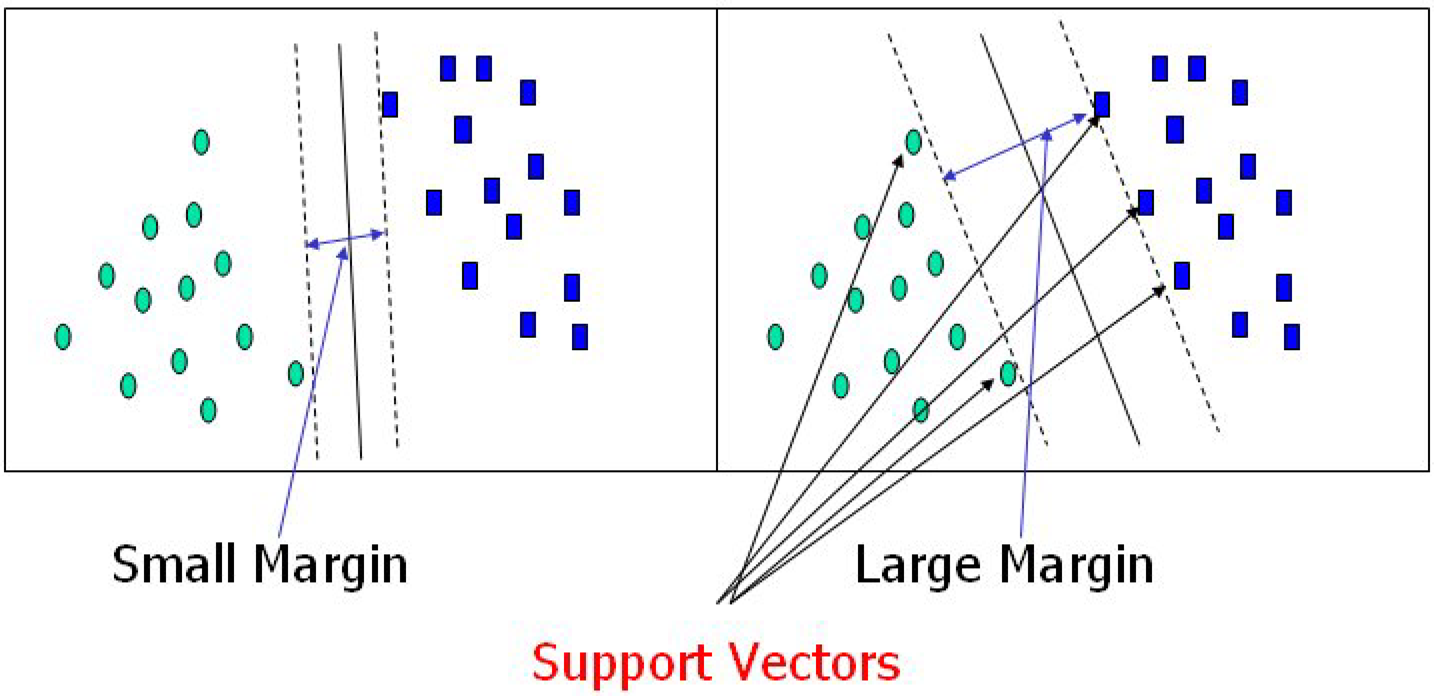
"Optimal" linear classifier:

Maximizing the margin



Support vector machine (SVM)

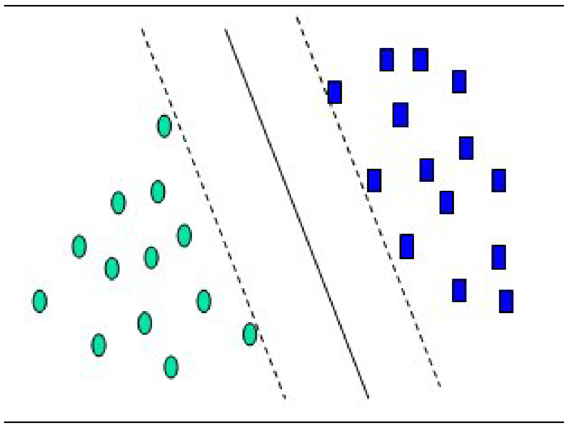
"Optimal" linear classifier



Support vector machine (SVM)

Training algorithm: Discrete optimization

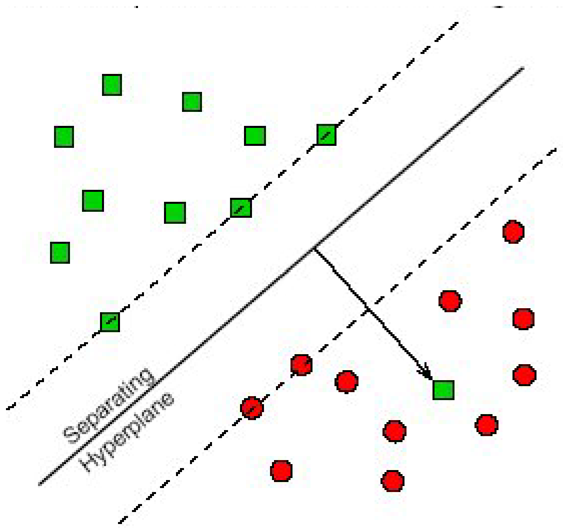
- many versions



Drawbacks:

- very sensitive to outliers and noise
- does not consider the shape and the size of the training set

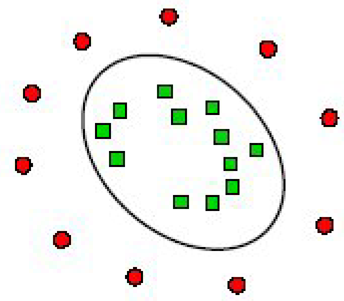
Admitting training errors



Cost function penalizes the errors

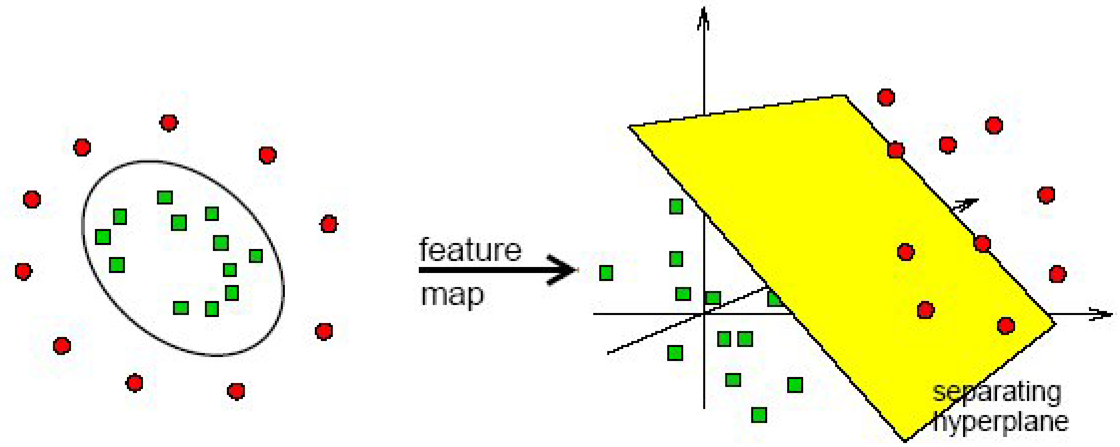
Attn: Definition of weights

SVM for linearly non-separable classes



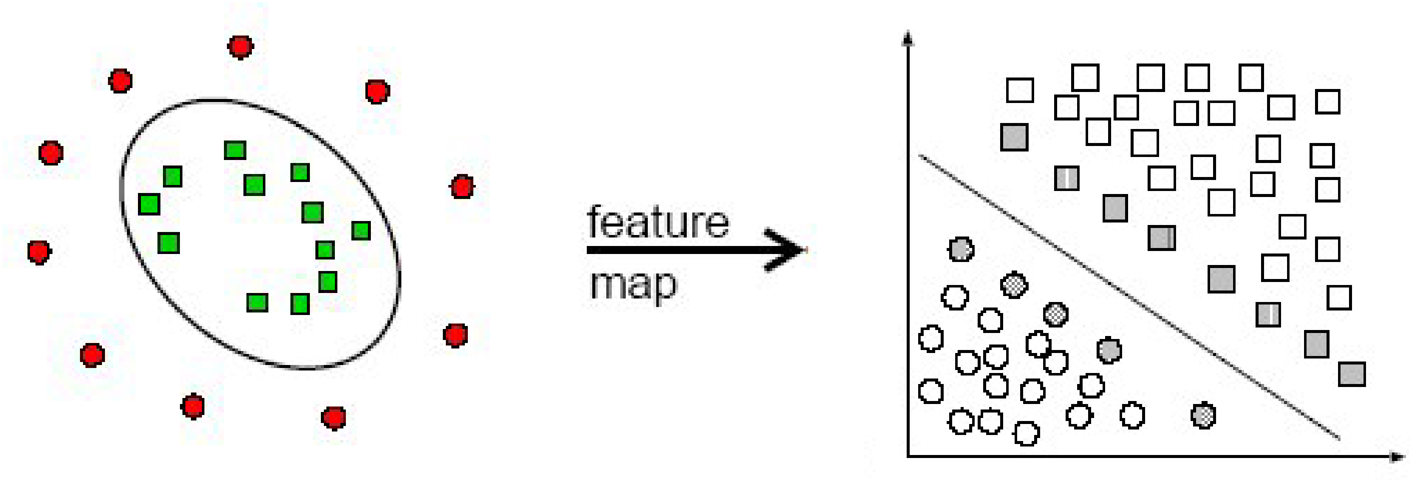
How to make them linearly separable?

Increasing the number of features



Attn: “Curse of dimensionality”

Mapping the features into another space
"The kernel trick"



Attn: "Overtraining/undertraining"

Bayesian classifier

Assumption: feature values are random variables

Statistic classifier, the decision is probabilistic

It is based on the **Bayes rule**

The Bayes rule

Class-conditional probability

A priori probability

A posteriori probability

$$P(\omega_j|\mathbf{x}) = \frac{p(\mathbf{x}|\omega_j)P(\omega_j)}{p(\mathbf{x})},$$

Total probability

$$p(\mathbf{x}) = \sum_{j=1}^{c} p(\mathbf{x}|\omega_j)P(\omega_j).$$

Bayesian classifier

Main idea: maximize posterior probability

$$P(\omega_j|\mathbf{x})$$

Since it is hard to do directly, we rather maximize

$$p(\mathbf{x}|\omega_j)P(\omega_j)$$

In case of equal priors, we maximize only

$$p(\mathbf{x}|\omega_j)$$

Equivalent formulation in terms of discriminat functions

$$g_i(\mathbf{x}) = P(\omega_i|\mathbf{x}) = \frac{p(\mathbf{x}|\omega_i)P(\omega_i)}{\sum_{j=1}^{c} p(\mathbf{x}|\omega_j)P(\omega_j)}$$

$$g_i(\mathbf{x}) = p(\mathbf{x}|\omega_i)P(\omega_i)$$

$$g_i(\mathbf{x}) = \ln p(\mathbf{x}|\omega_i) + \ln P(\omega_i),$$

How to estimate $p(\mathbf{x}|\omega_j)P(\omega_j)$?

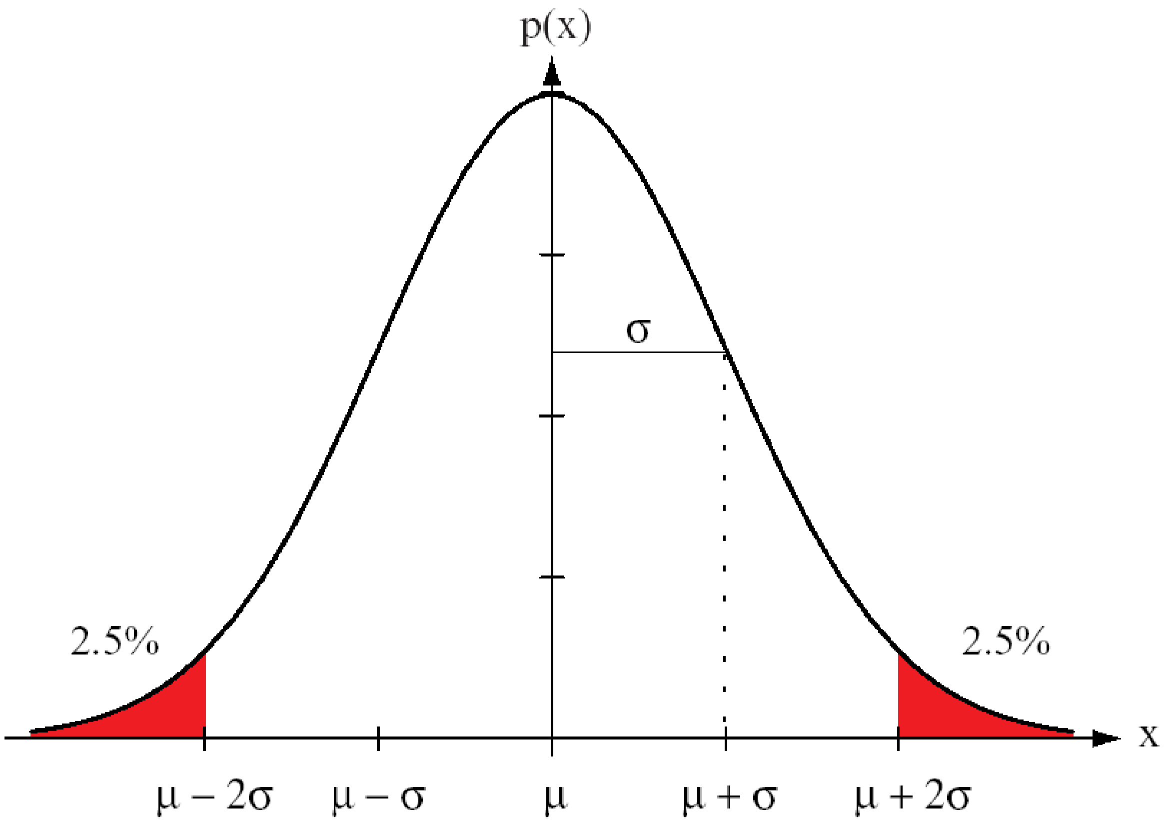
$P(\omega_j)$

- From the case studies performed before (OCR, speech recognition)
- From the occurence in the training set
- Assumption of equal priors

$p(\mathbf{x}|\omega_j)$

- Parametric estimate (assuming pdf is of a known form, e.g. Gaussian)
- Non-parametric estimate (pdf is unknown or too complex)

Parametric estimate of Gaussian $p(\mathbf{x}|\omega_j)$.



$$\hat{\mu} = \frac{1}{n} \sum_{k=1}^{n} x_k$$

$$\hat{\sigma}^2 = \frac{1}{n} \sum_{k=1}^{n} (x_k - \hat{\mu})^2$$

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right]$$

A note about the ML estimator

$$f(x; \theta_1, \theta_2, ..., \theta_k)$$

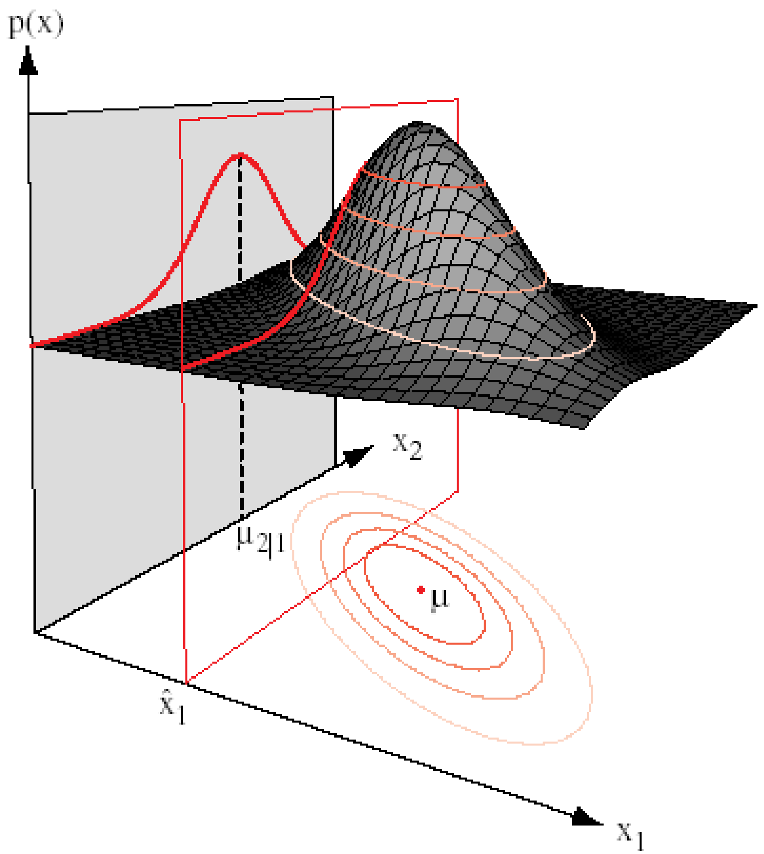
$$L(x_1, x_2, ..., x_N | \theta_1, \theta_2, ..., \theta_k) = L = \prod_{i=1}^{N} f(x_i; \theta_1, \theta_2, ..., \theta_k)$$

$$i = 1, 2, ..., N$$

$$\Lambda = \ln L = \sum_{i=1}^{N} \ln f(x_i; \theta_1, \theta_2, ..., \theta_k)$$

$$\frac{\partial(\Lambda)}{\partial \theta_j} = 0, \quad j = 1, 2, ..., k$$

d-dimensional Gaussian pdf

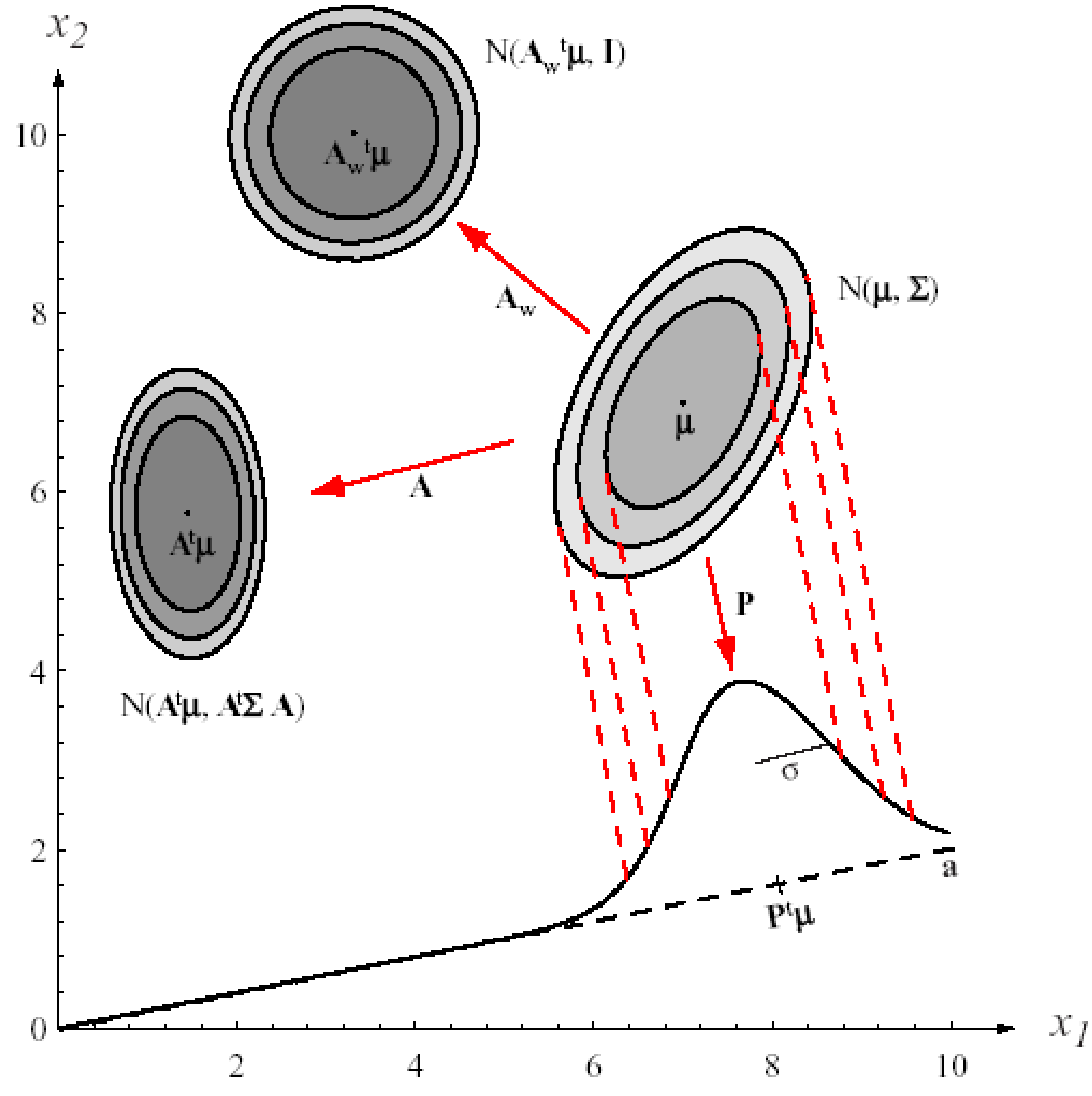


$$\hat{\mu} = \frac{1}{n}\sum_{k=1}^{n}\mathbf{x}_k$$

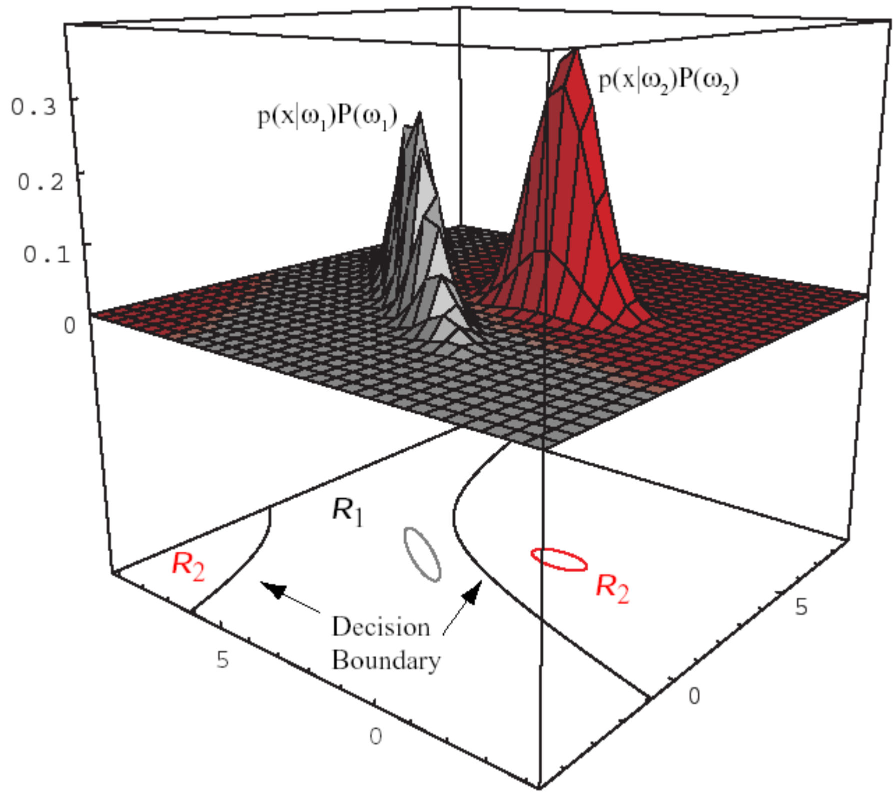
$$\widehat{\mathbf{\Sigma}} = \frac{1}{n}\sum_{k=1}^{n}(\mathbf{x}_k - \hat{\mu})(\mathbf{x}_k - \hat{\mu})^t.$$

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\mathbf{\Sigma}|^{1/2}}\exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^t\mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right]$$

The role of covariance matrix



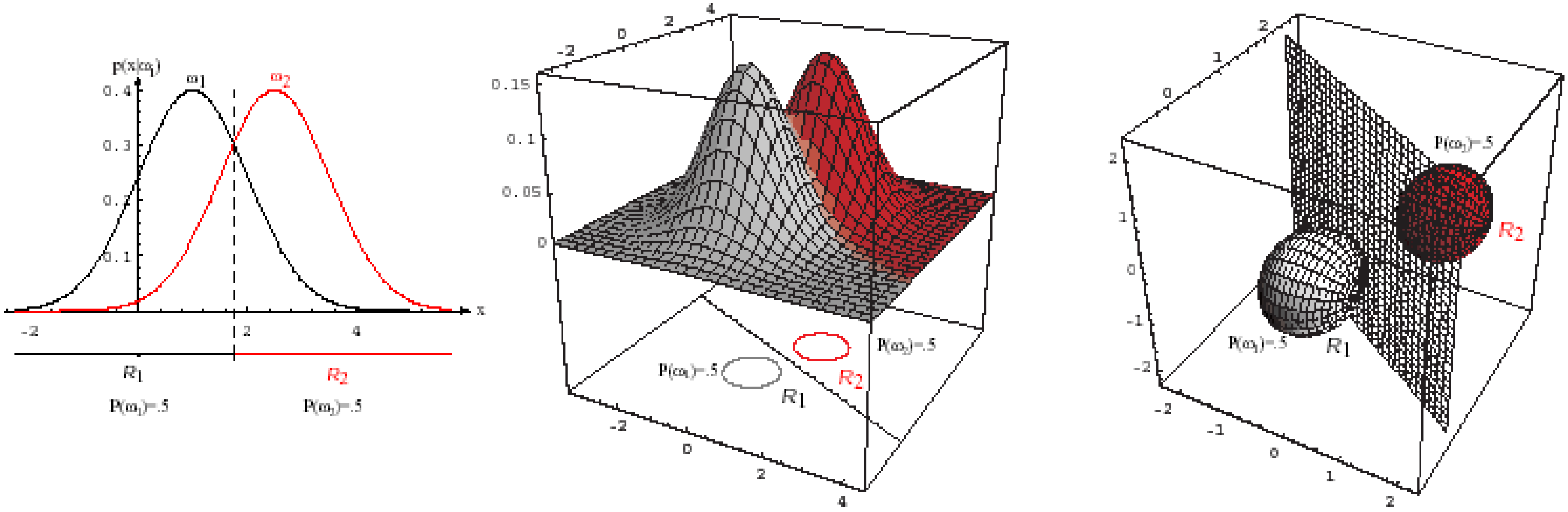
Two-class Gaussian case in 2D



$$g_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^t\boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i) - \frac{d}{2}\ln 2\pi - \frac{1}{2}\ln|\boldsymbol{\Sigma}_i| + \ln P(\omega_i).$$

Classification = comparison of two Gaussians

Two-class Gaussian case – Equal cov. mat.



$$g_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^t\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_i) + \ln P(\omega_i)$$

Linear decision boundary

Equal priors $P(\omega_j)$

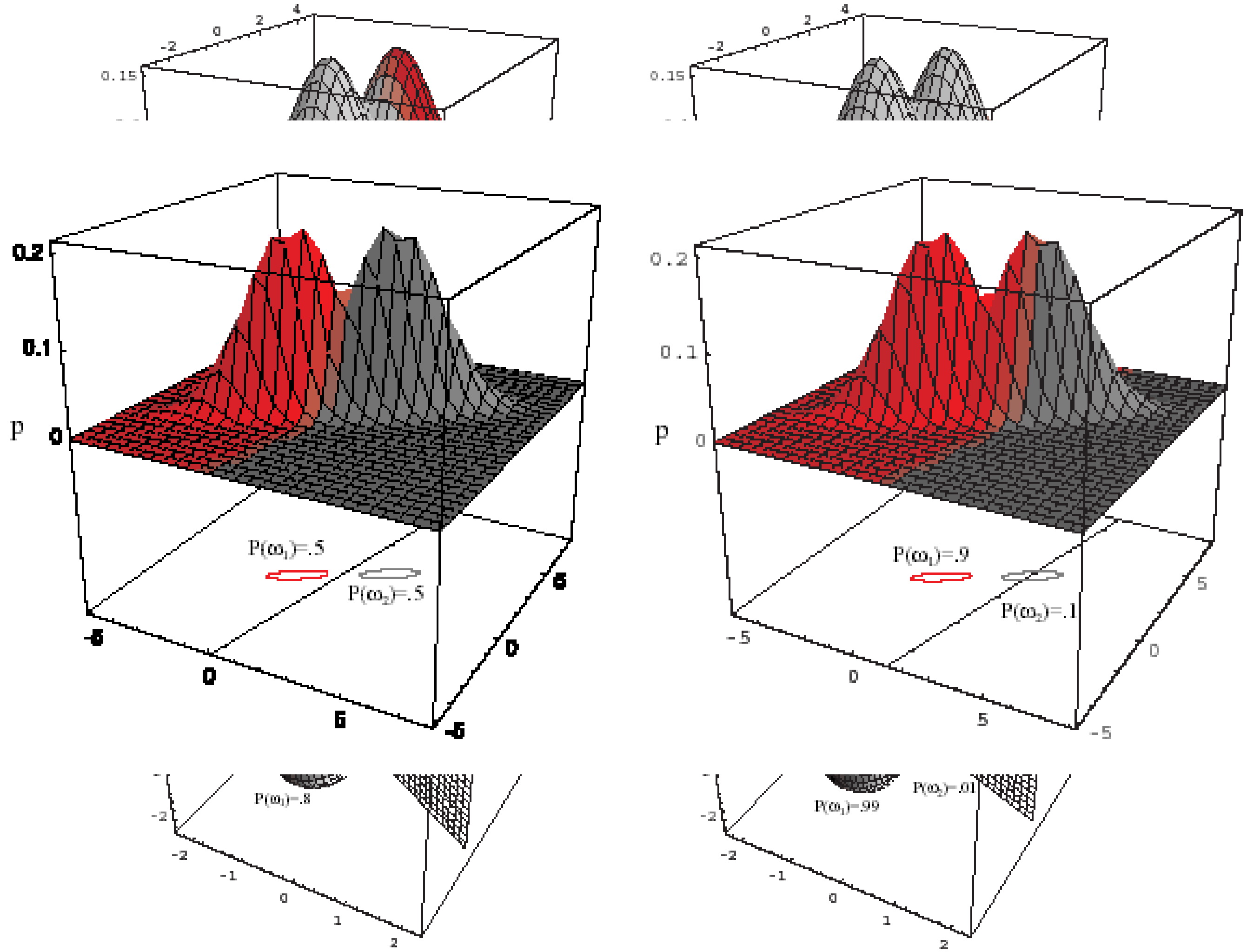
max $g_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_i)^t \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}_i)$.

min $(\mathbf{x} - \boldsymbol{\mu}_i)^t \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}_i)$.

Classification by minimum Mahalanobis distance

If the cov. mat. is diagonal with equal variances then we get "standard" minimum distance rule

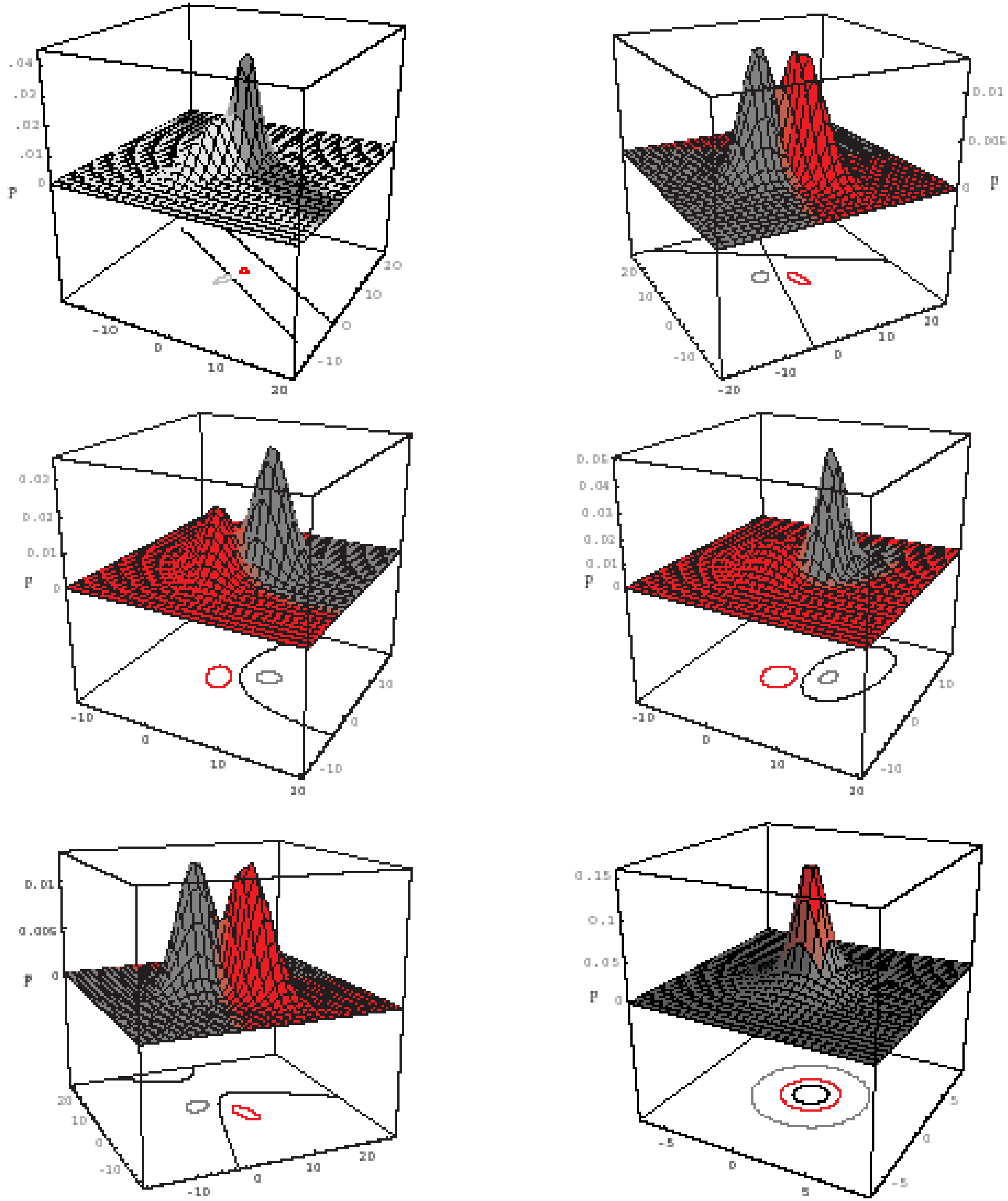
Non-equal priors $P(\omega_j)$



Linear decision boundary still preserved

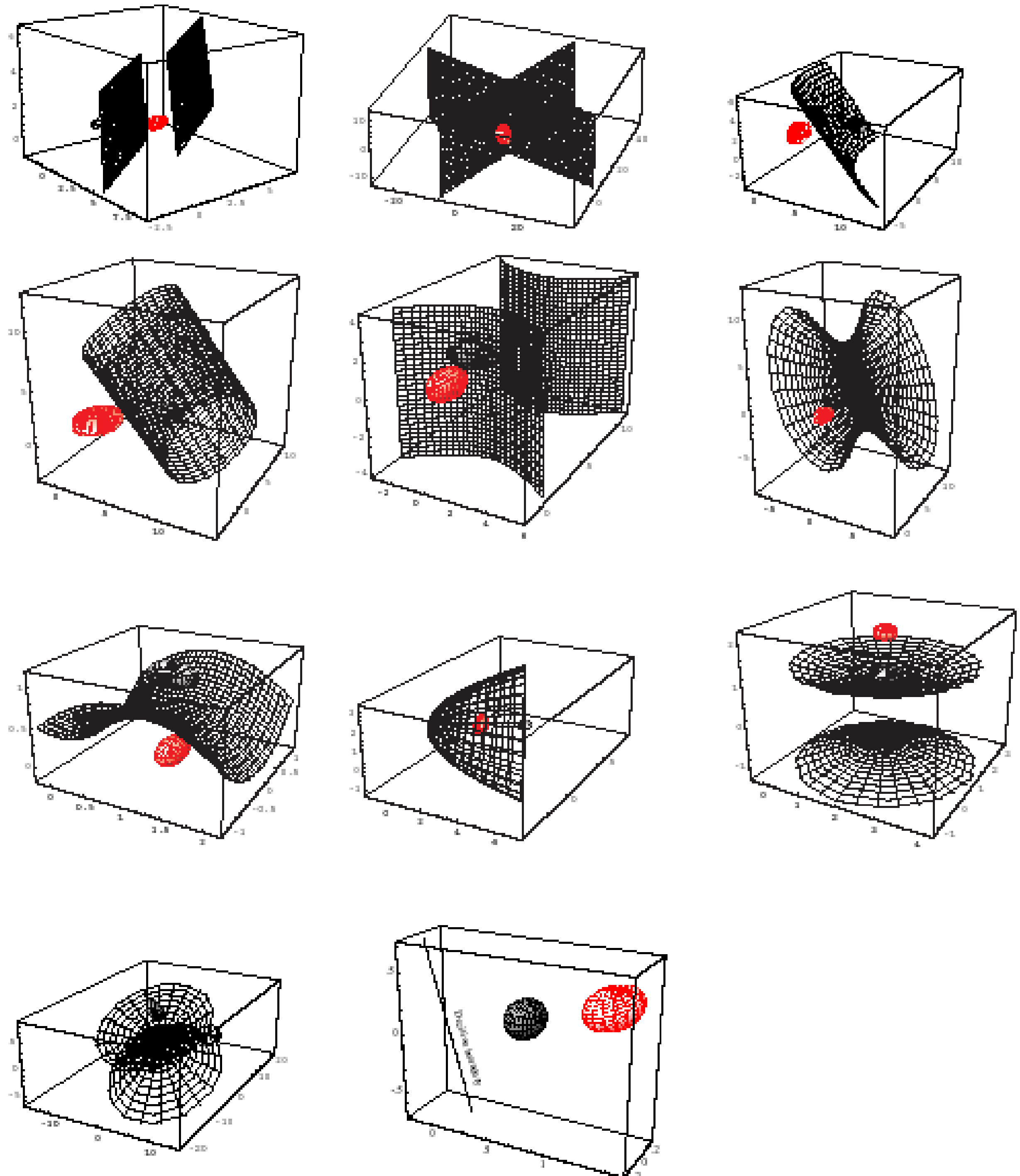
General G case in 2D

Decision boundary is a hyperquadric

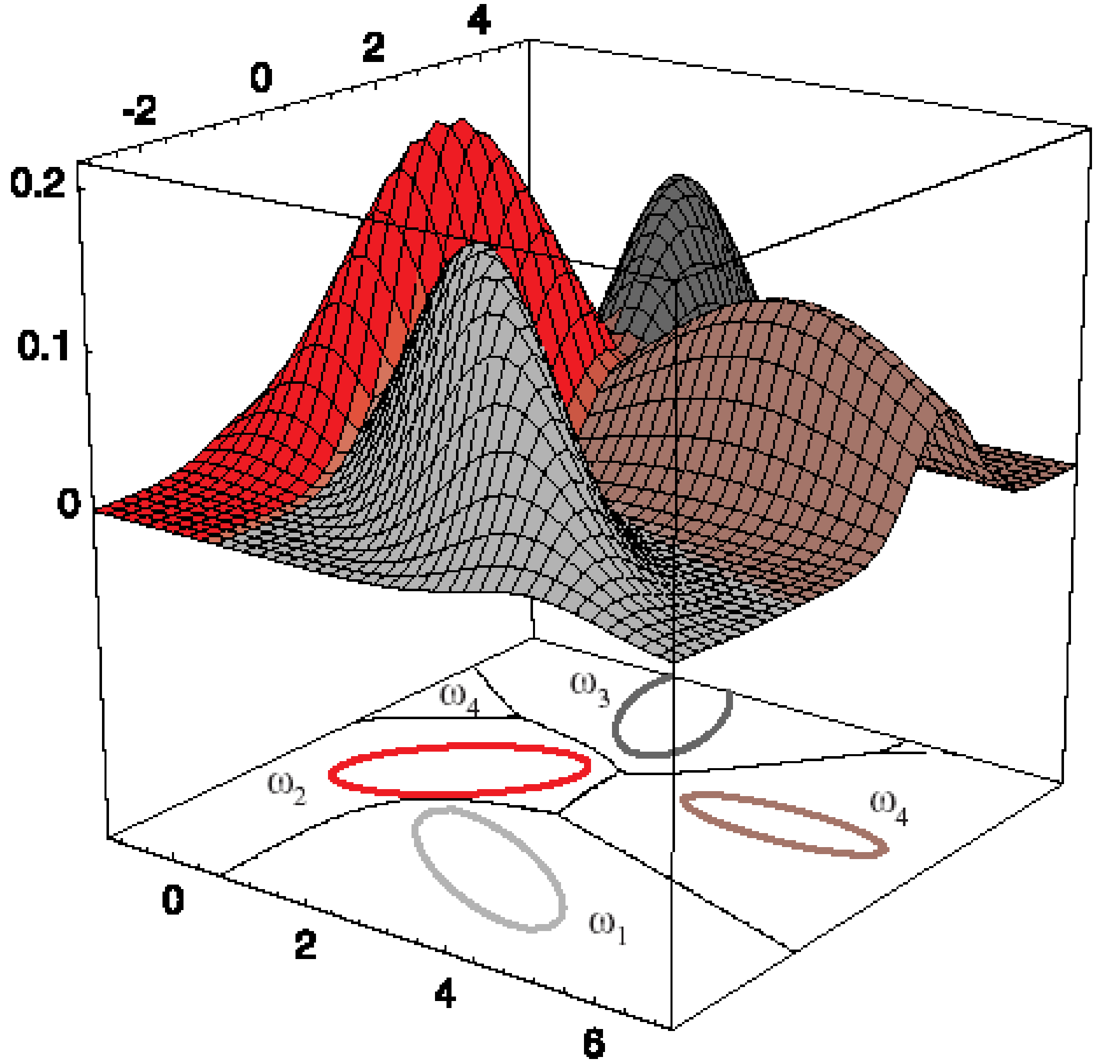


General G case in 3D

Decision boundary is a hyperquadric



More classes, Gaussian case in 2D



How to test the normality of the class-conditional distributions?

- Pearson's chi-square test (goodness-of-fit test)

$$X^2 = \sum \frac{(\text{observed - expected})^2}{\text{expected}}$$

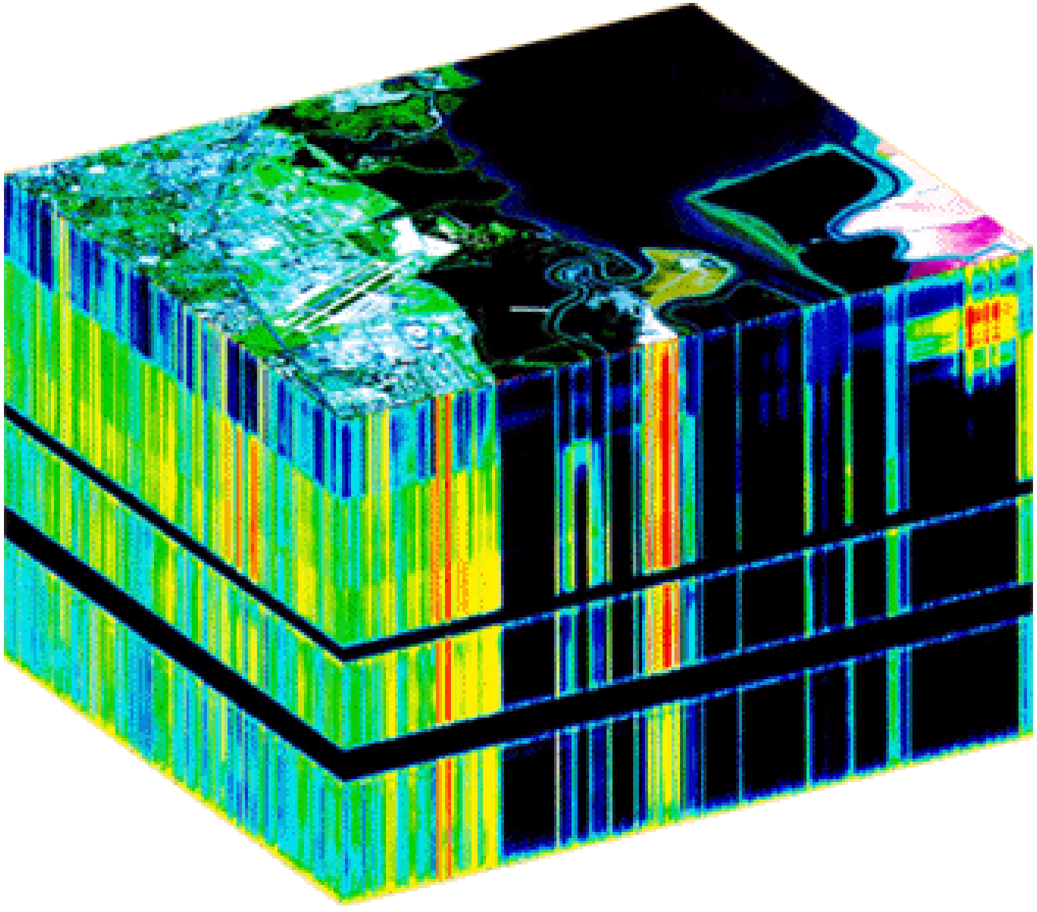
- Moment-based tests

- Visual assessment

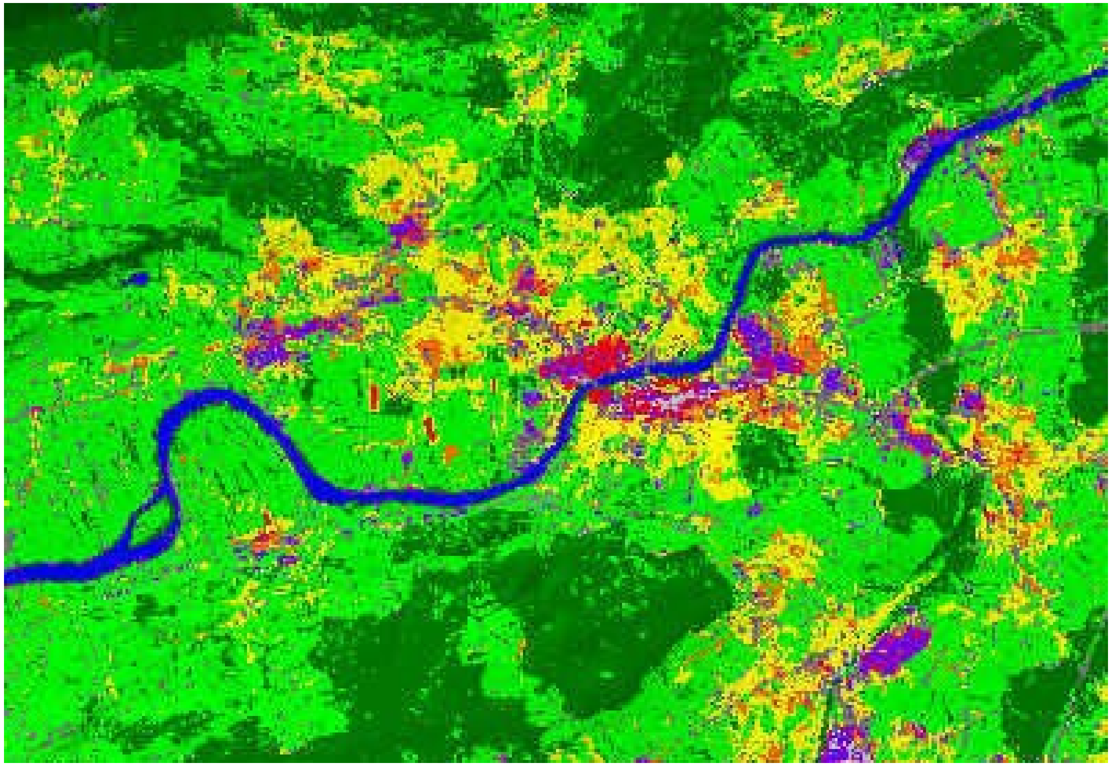
Applications of Bayesian classifier in multispectral remote sensing

- Objects = pixels
- Features = pixel values in the spectral bands (from 4 to several hundreds)
- Training set – selected manually by means of thematic maps (GIS), and on-site observation
- Number of classes – typically from 2 to 16

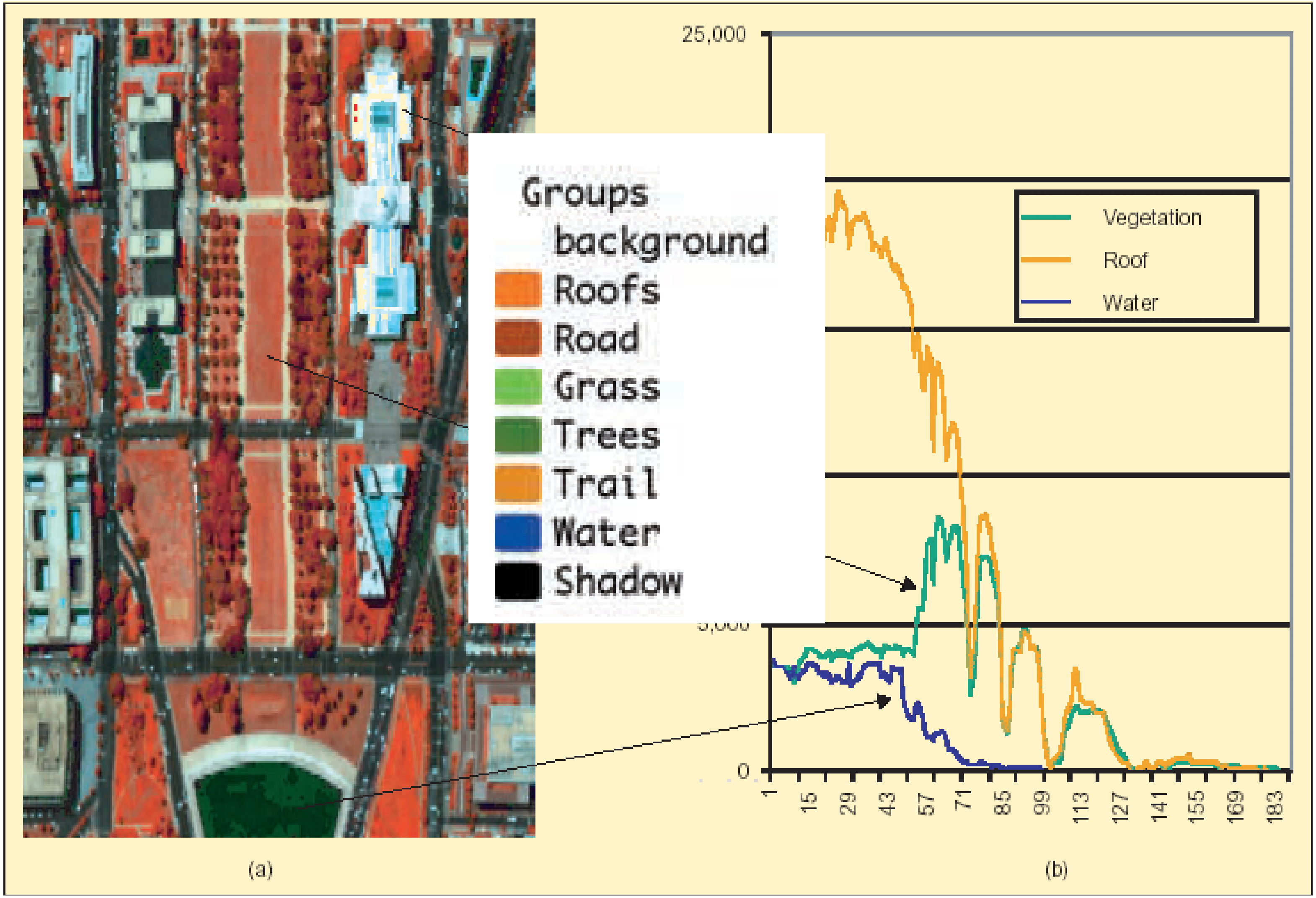
Applications of Bayesian classifier in multispectral remote sensing

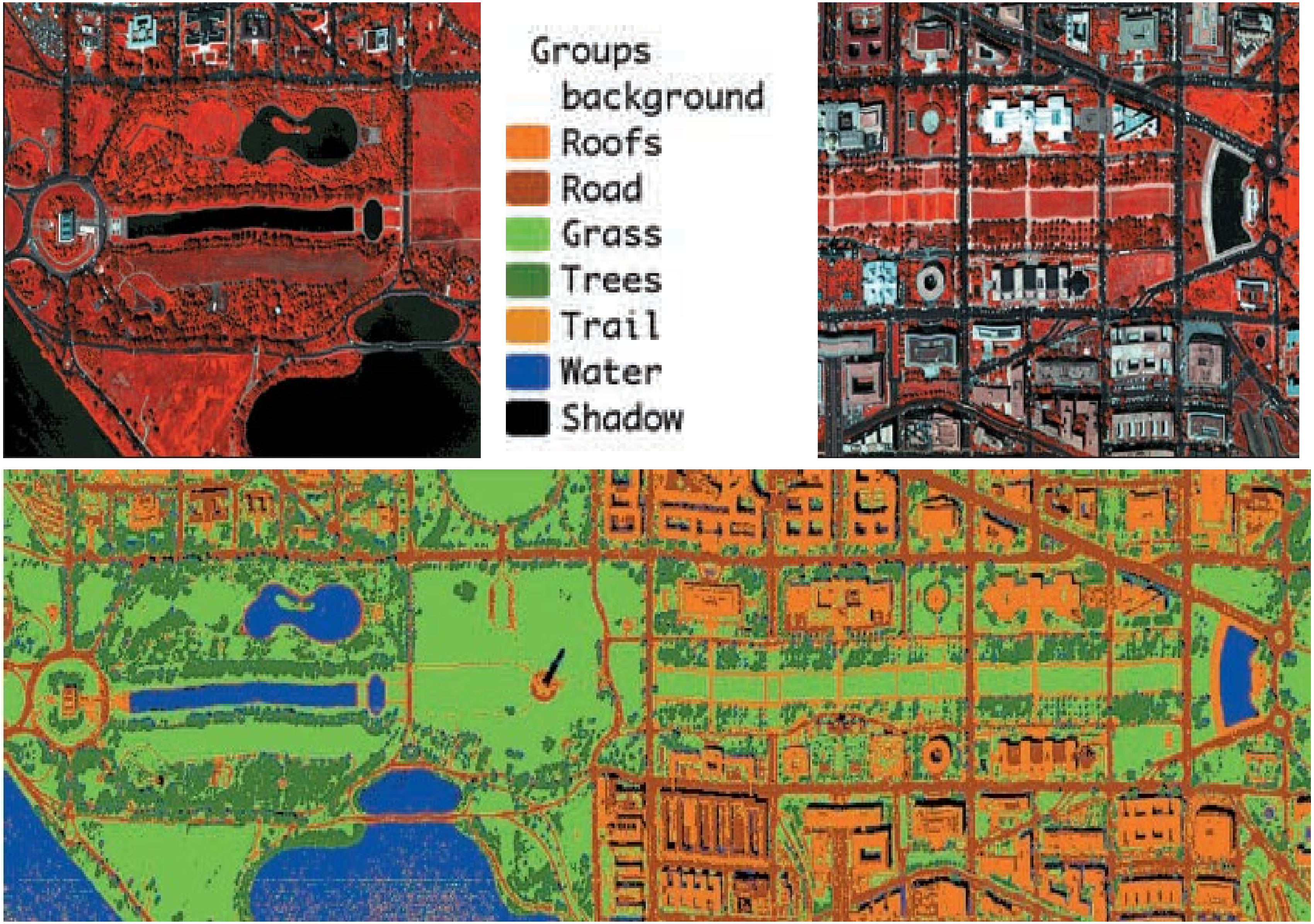


Applications of Bayesian classifier in multispectral remote sensing

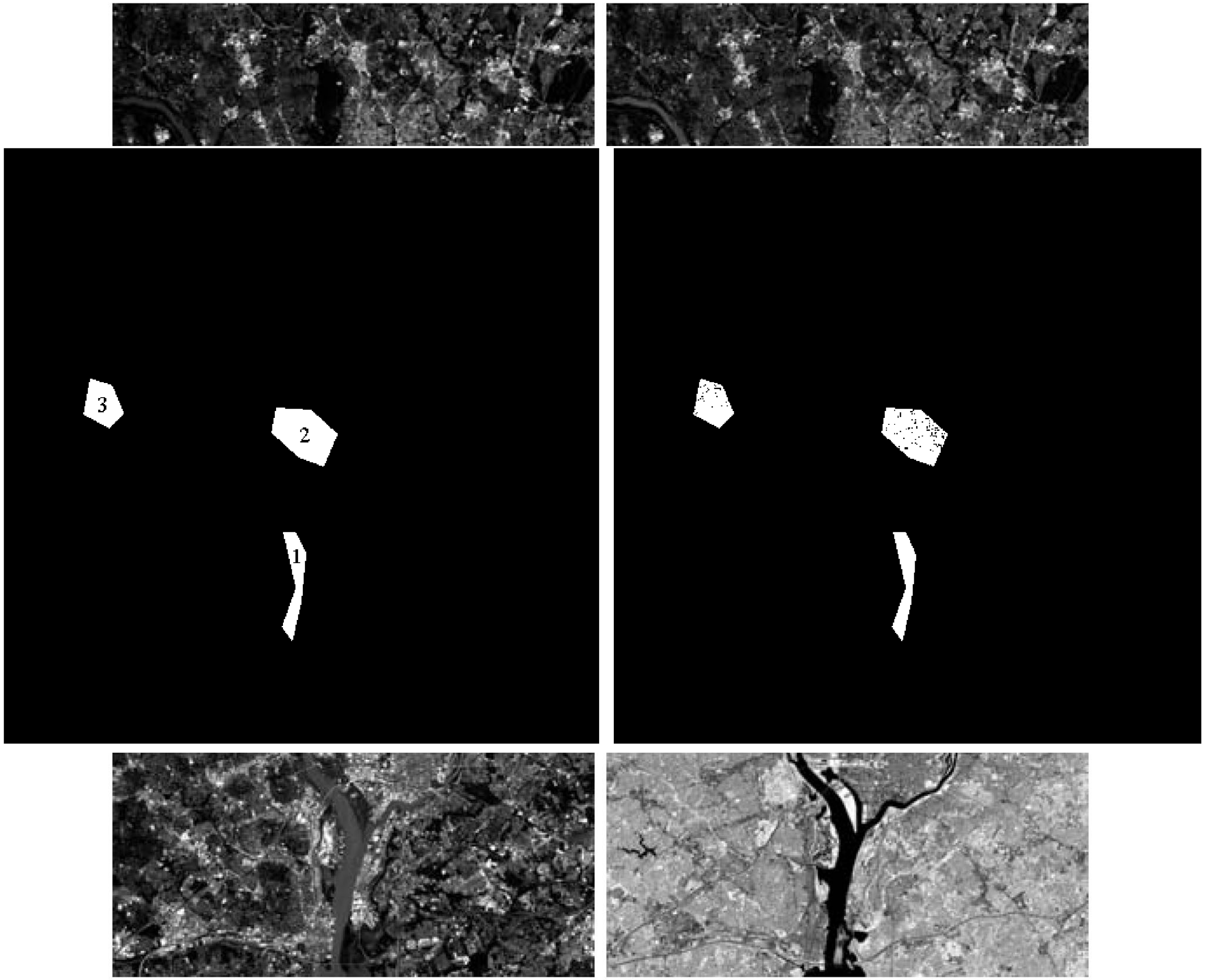


The Mall, Washington D.C, aerial HS image

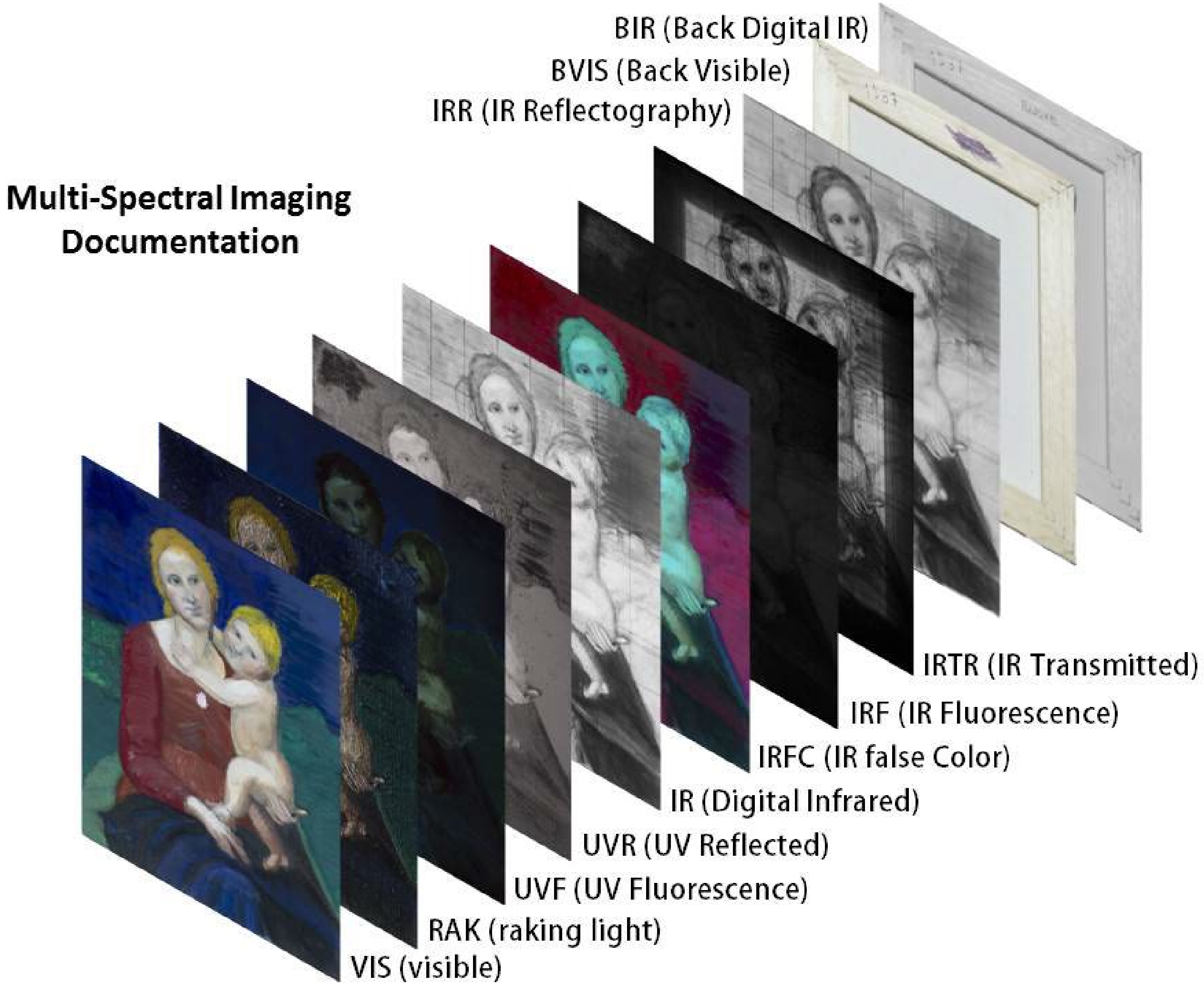



Groups
background
Roofs
Road
Grass
Trees
Trail
Water
Shadow

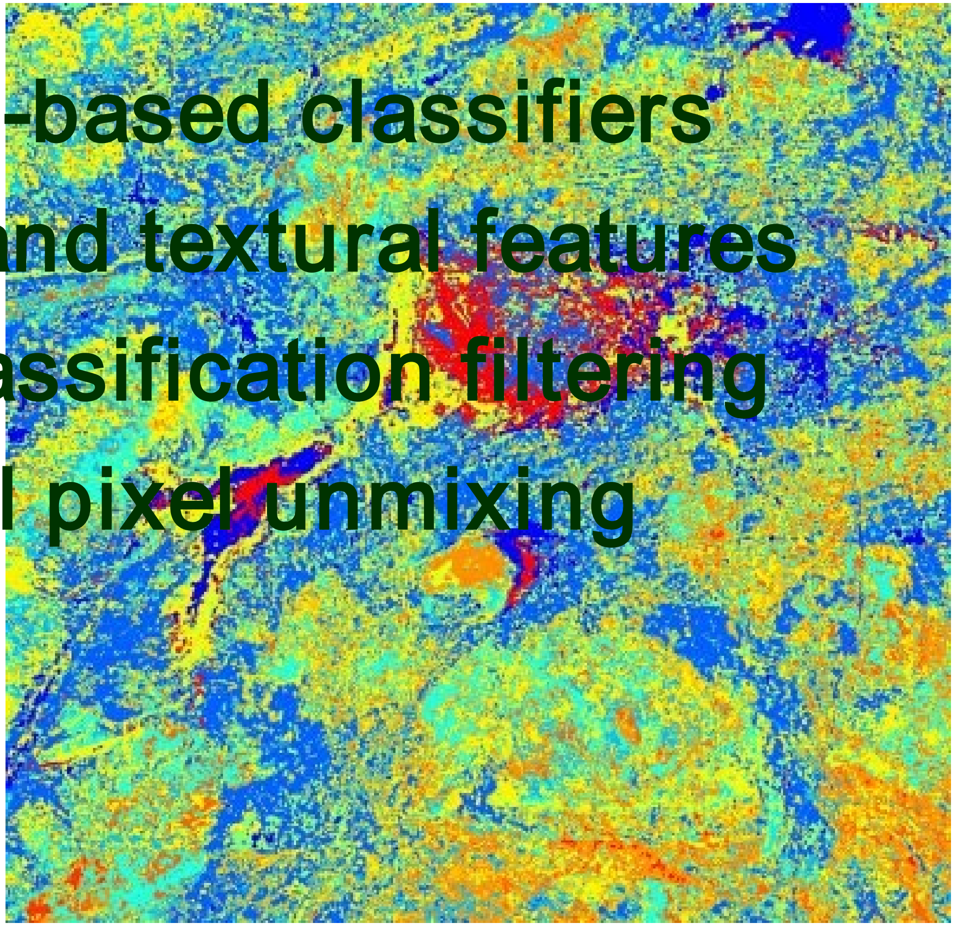
Satellite MS image – Training set selection



Applications of Bayesian classifier in art



Other classification methods in RS



- Context-based classifiers
- Shape and textural features
- Post-classification filtering
- Spectral pixel unmixing

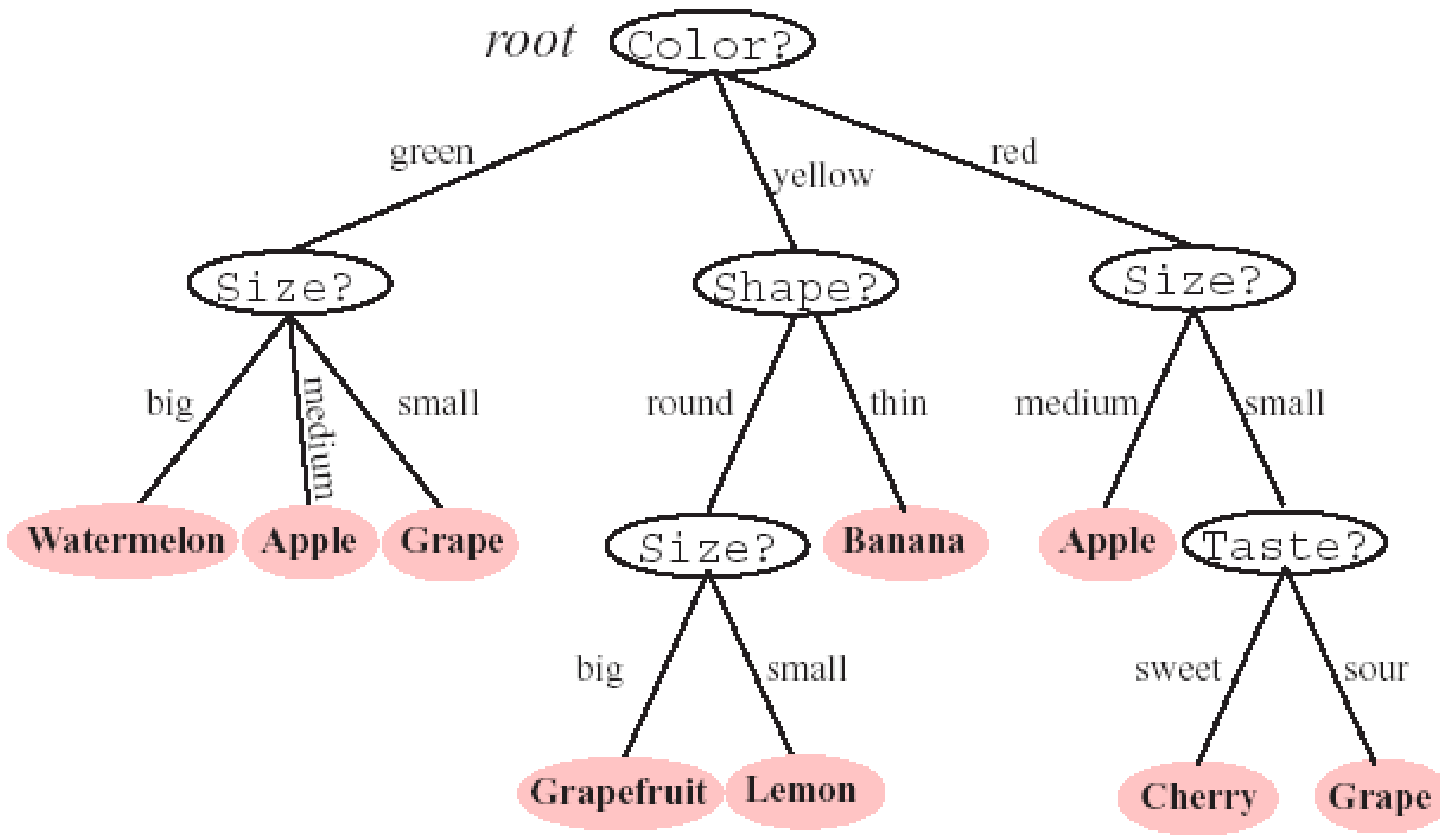
Non-metric classifiers

Typically for “YES – NO” features

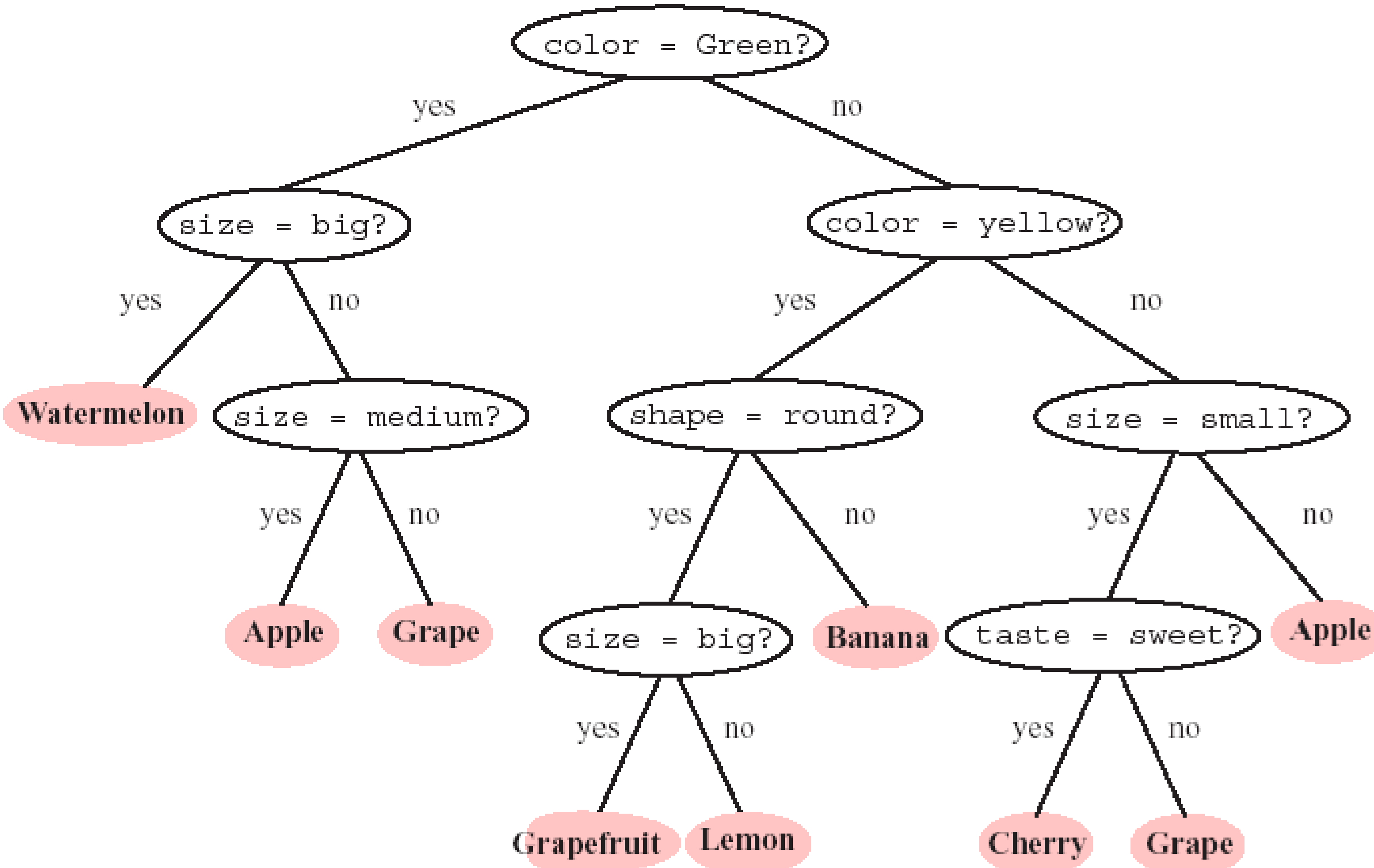
Feature metric is not explicitly defined

Decision trees

General decision tree



Binary decision tree

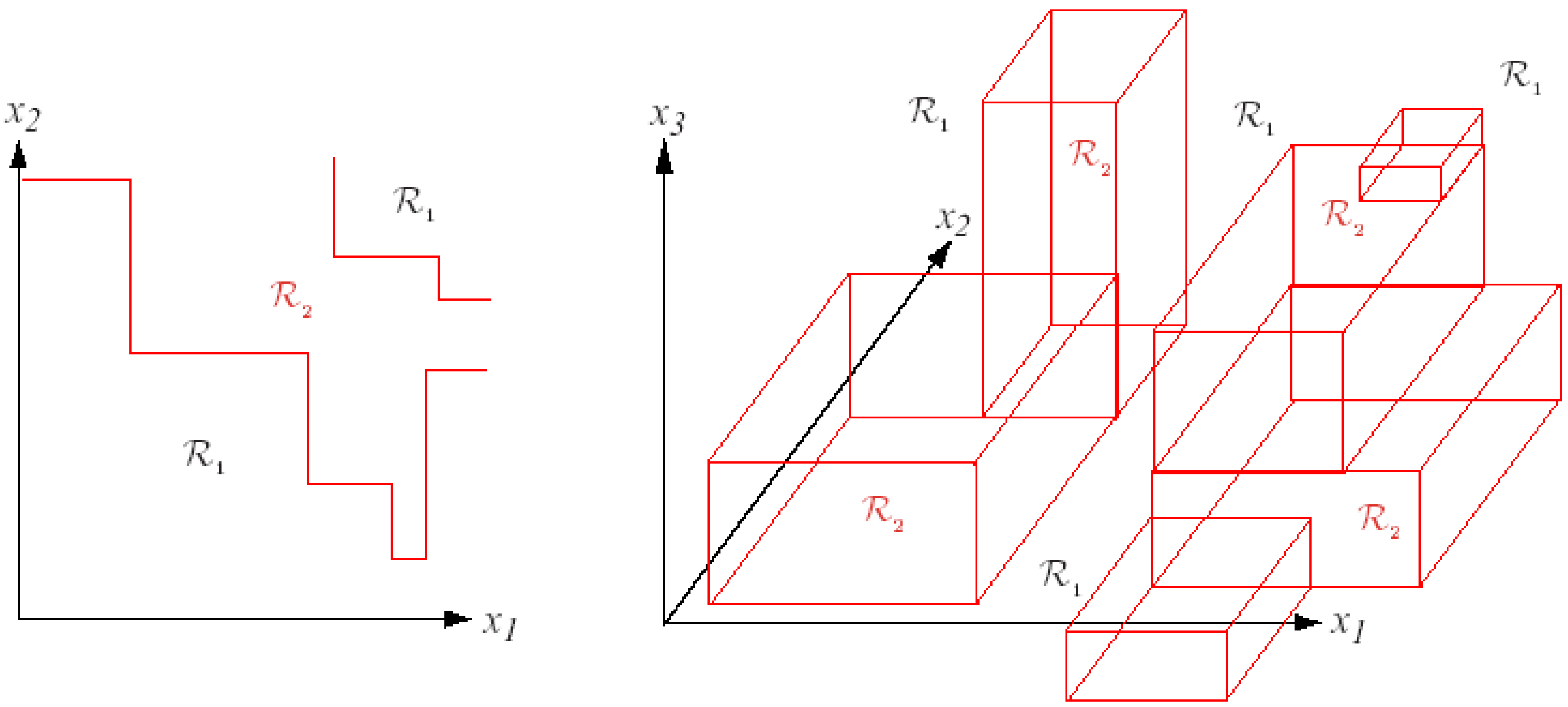


Any decision tree can be replaced by a binary tree

Real-valued features

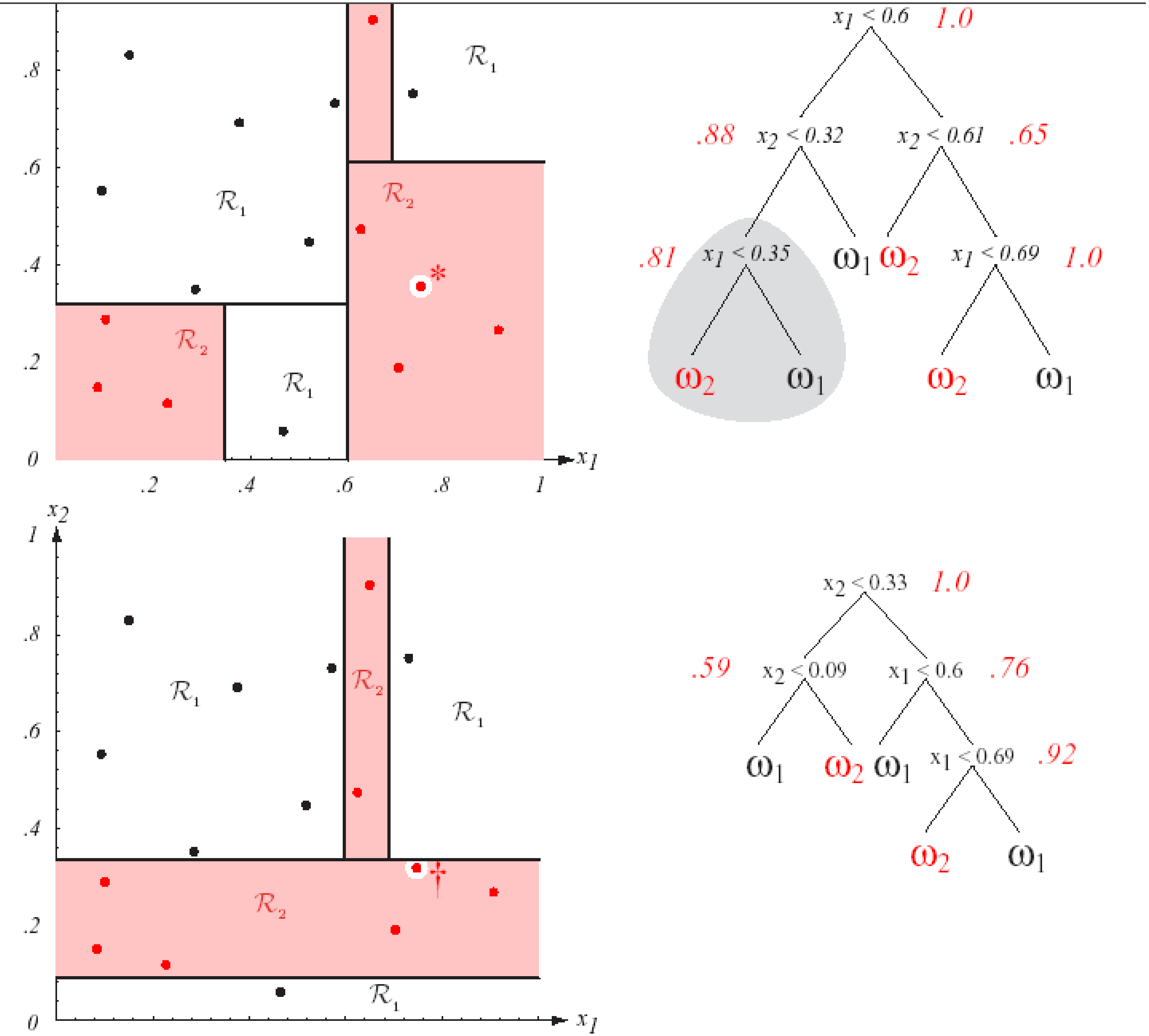
- Node decisions are in form of inequalities
- Training = setting their parameters
- Simple inequalities → stepwise decision boundary

Stepwise decision boundary

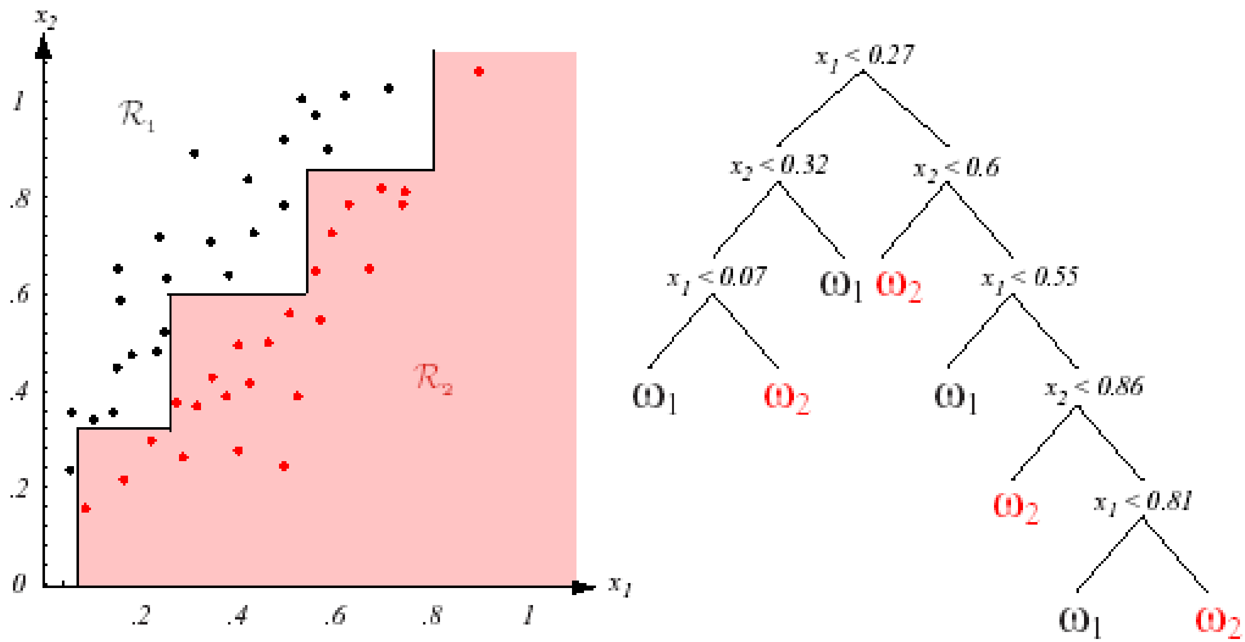


Real-valued features

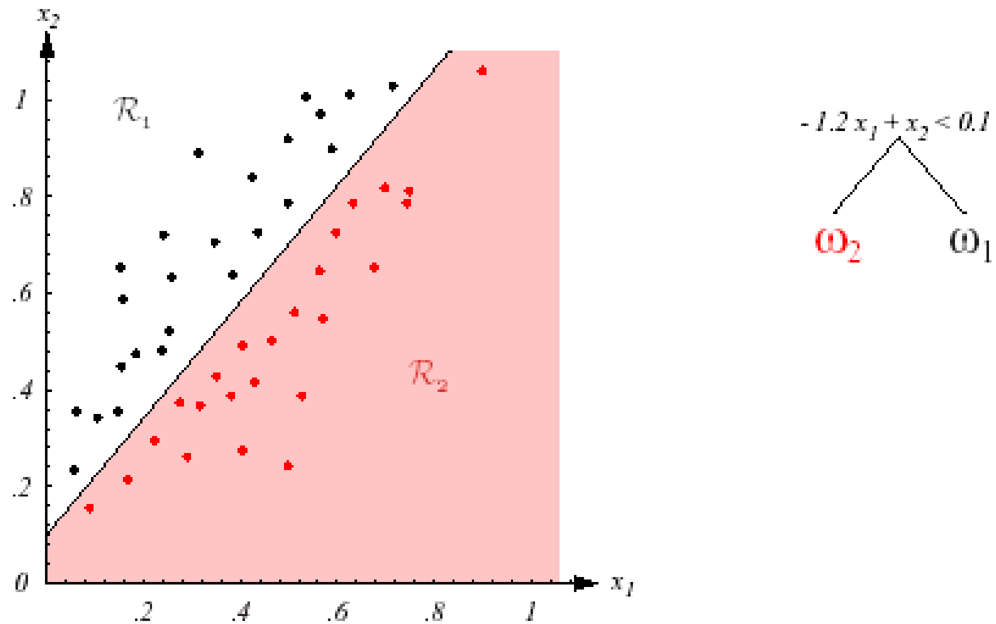
The tree structure and the form of inequalities influence both performance and speed.



x1 < 0.6 1.0
.88 x2 < 0.32
x2 < 0.61 .65
.81 x1 < 0.35
ω1 ω2
x1 < 0.69 1.0
ω2 ω1 ω2 ω1
x2 < 0.33 1.0
.59 x2 < 0.09
x1 < 0.6 .76
ω1 ω2 ω1
x1 < 0.69 .92
ω2 ω1



$x_1 < 0.27$
$x_2 < 0.32$
$x_2 < 0.6$
$x_1 < 0.07$
ω_1
ω_2
$x_1 < 0.55$
ω_1
ω_2
ω_1
$x_2 < 0.86$
ω_2
$x_1 < 0.81$
ω_1
ω_2
$\mathcal{R}_1$
$\mathcal{R}_2$



$-1.2\,x_1 + x_2 < 0.1$
ω_2
ω_1
$\mathcal{R}_1$
$\mathcal{R}_2$

Classification performance

- **How to evaluate the performance of the classifiers?**

  - evaluation on the training set (optimistic error estimate)

  - evaluation on the test set. Evaluation by overall success rate is misleading, different errors may have different significance. One should use the *confusion table.*

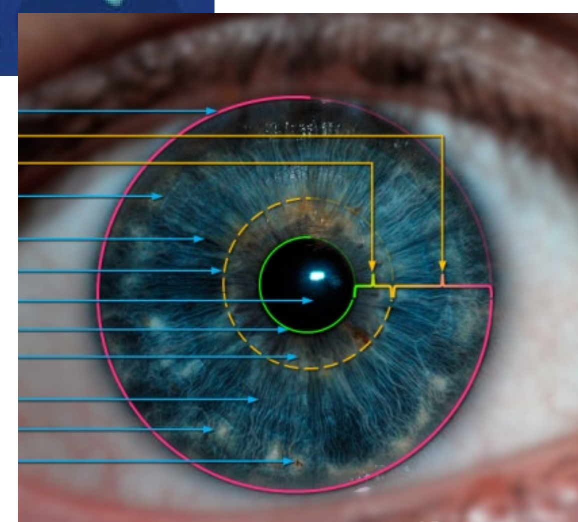
Classification performance

- **How to increase the performance?**
  - other features
  - more features (dangerous – curse of dimensionality!)
  - other (larger, better) training sets
  - other parametric models
  - other classifiers
  - **combining different classifiers**

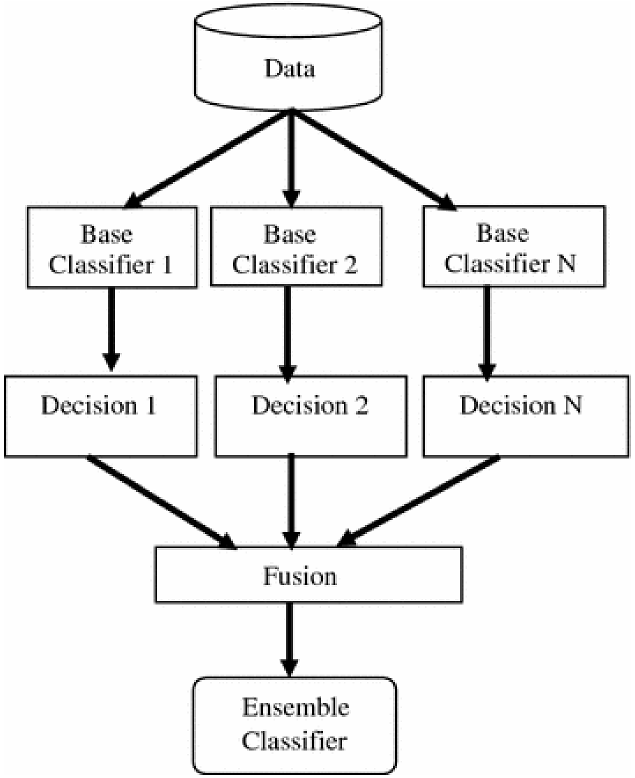
Combining classifiers

- Several *independent* classifiers
- Averaging of noisy results
- Suppression of extremes
- No guarantee of improvement over the best classifier

Combining classifiers in biometrics



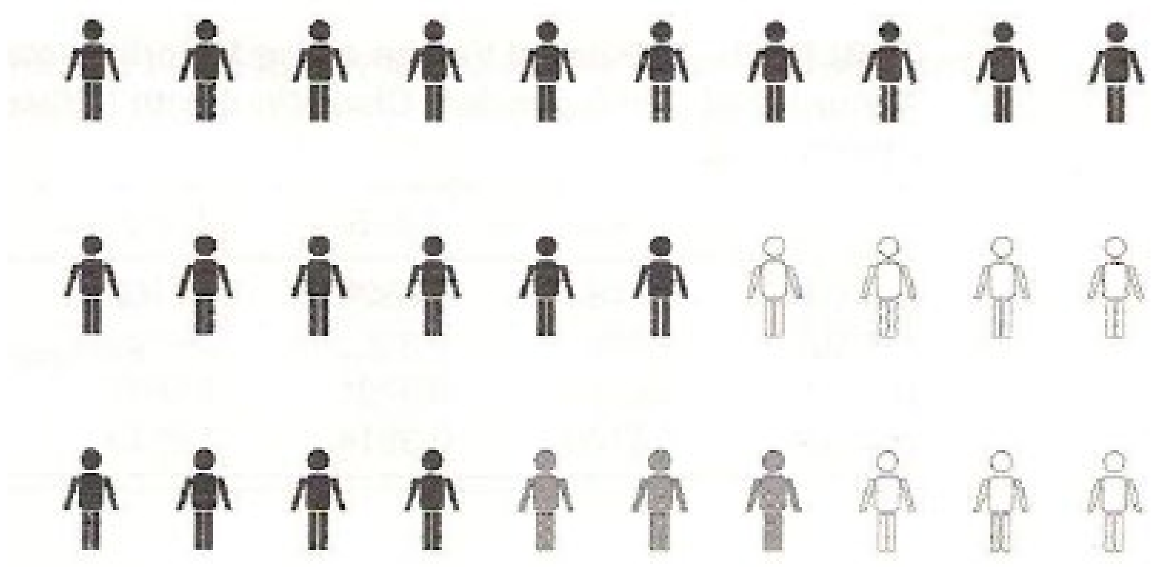
Combining deterministic decisions: voting



$$P_{maj} = \sum_{m=\lfloor L/2 \rfloor + 1}^{L} \binom{L}{m} p^m (1-p)^{L-m}$$

$$P > p \quad \text{iff} \quad p > 0.5$$

Combining deterministic classifiers: voting



Weighted voting: incorporating the expert knowledge

Majority vote – probability of success

Většinové hlasování :

$$P_{maj} = \sum_{m=\lfloor L/2 \rfloor +1}^{L} \binom{L}{m} p^m (1-p)^{L-m}$$

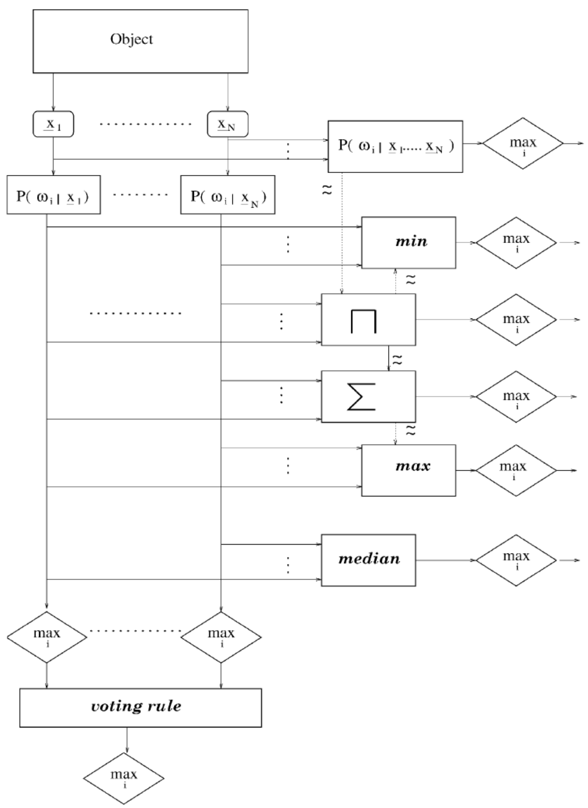
kde L je počet **nezávislých** klasifikátorů a p je jejich úspěšnost.

	$L=3$	$L=5$	$L=7$	$L=9$
$p=0.6$	0.6480	0.6826	0.7102	0.7334
$p=0.7$	0.7840	0.8369	0.8740	0.9012
$p=0.8$	0.8960	0.9421	0.9667	0.9804
$p=0.9$	0.9720	0.9914	0.9973	0.9991

Combining probabilistic classifiers

$$\max \quad p(\omega_i | x_1, \cdots, x_C)$$

$$p(\omega_i) \cdot p(x_1, \cdots, x_C | \omega_i) \qquad p(x_1, \cdots, x_C | \omega_i) = \prod_{j=1}^{C} p(x_j | \omega_i)$$



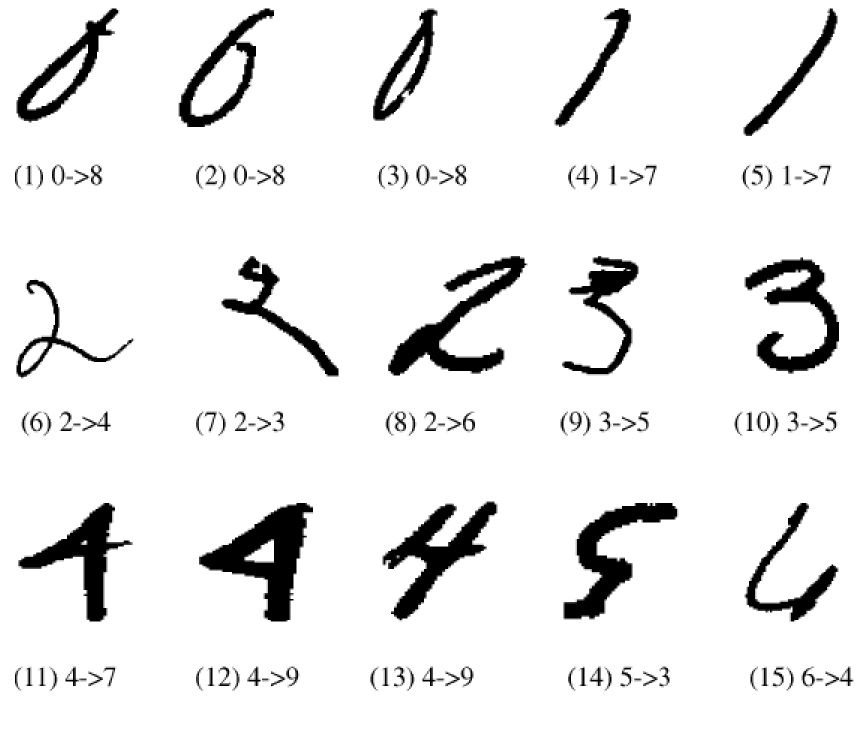
Handwritten digit recognition

TABLE 2
THE CLASSIFICATION RATE FOR EACH CLASSIFIER

Individual classifier	Classification rate %
Structural:	90.85
Gaussian:	93.93
Neural Net:	93.2
HMM:	94.77

TABLE 3
THE CLASSIFICATION RATE USING DIFFERENT COMBINING SCHEMES

Combining rule	Classification rate %
Majority Vote:	97.96
Sum rule:	98.05
Max rule:	93.93
Min rule:	86.00
Product rule:	84.69
Median rule:	98.19



Reprinted from J. Kittler et al., PAMI, 1998

Thank you !

Any questions ?