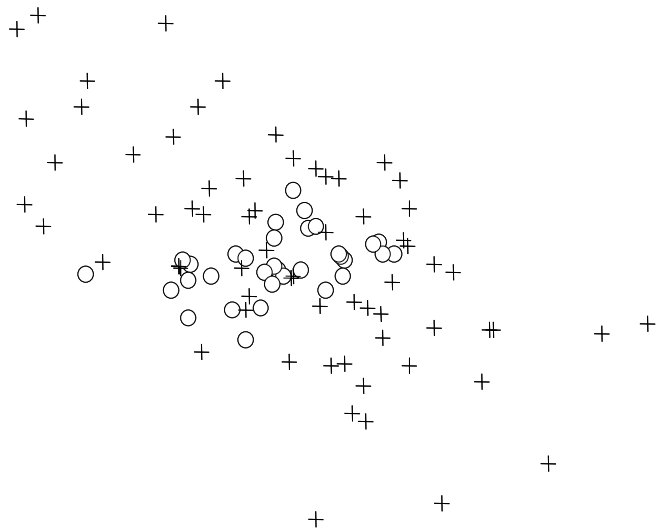
Pattern Recognition

Summary

- Motivation
- Introduction to Pattern Recognition
- Optimal Classifiers
- Learning

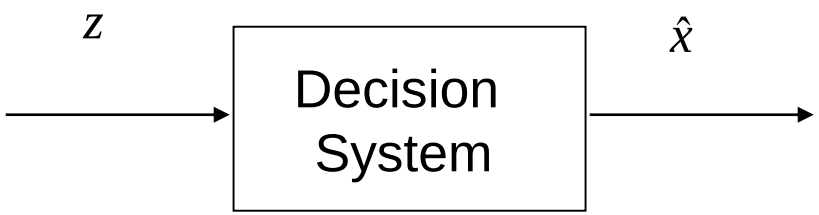
Challenge



How to decide if a pattern belongs to class 1 or 2 ?

Pattern Recognition

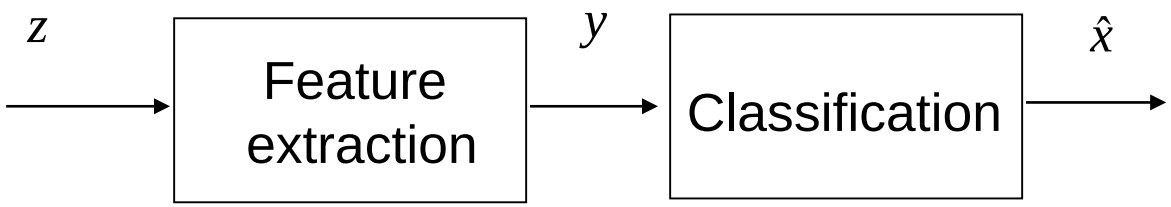
Main Problem: *classify an observation (signal, image, etc) in one of c admissible classes.*



z observed signal

$\hat{x}$ estimated class

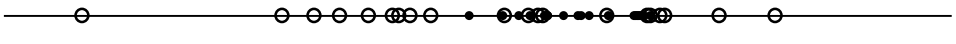
Classic Architecture



z	observed signal	
y	feature vector (pattern)	$y \in S$
$\hat{x}$	estimated class	$\hat{x} \in \{1,2,\ldots,c\}$

2 Populations

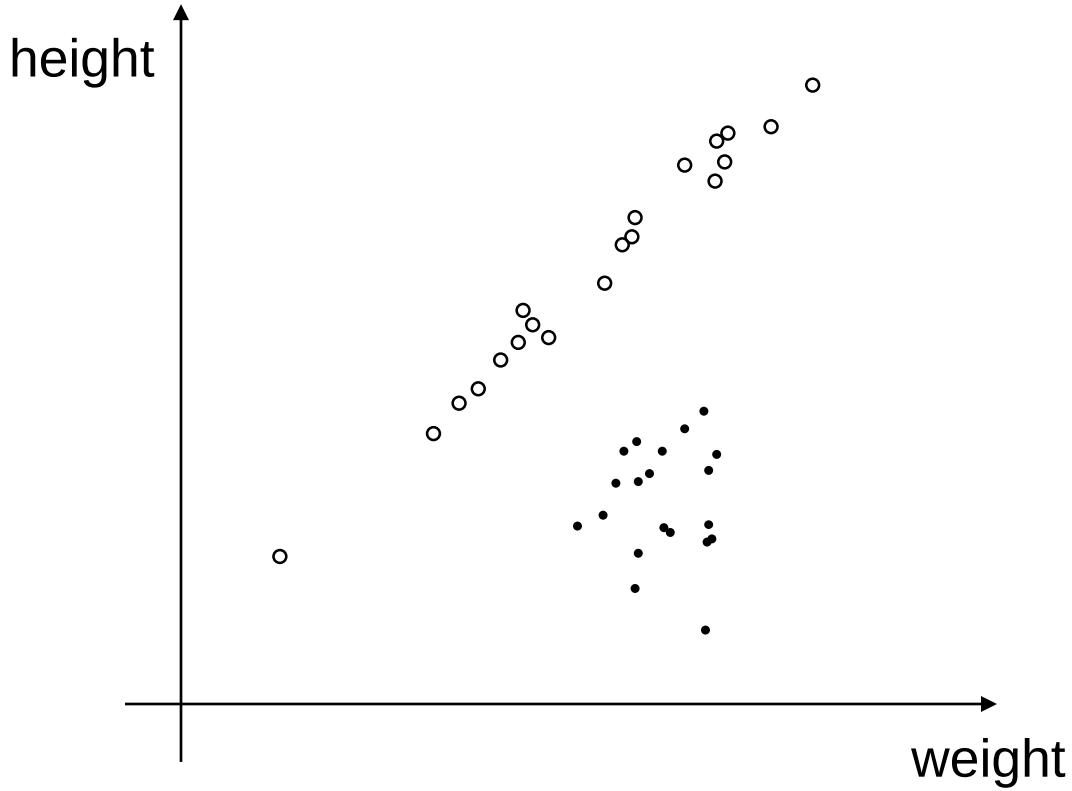
Problem: distinguish two populations of persons



weight

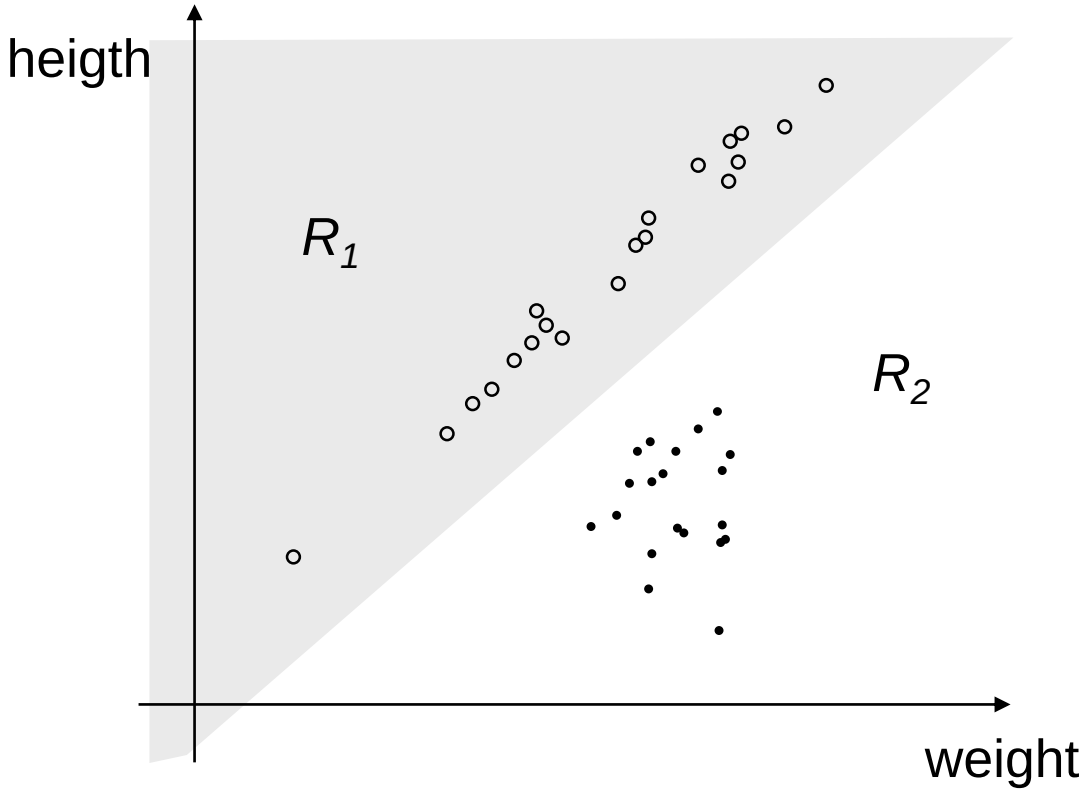
It is not possible to classify without errors!

2 Populations

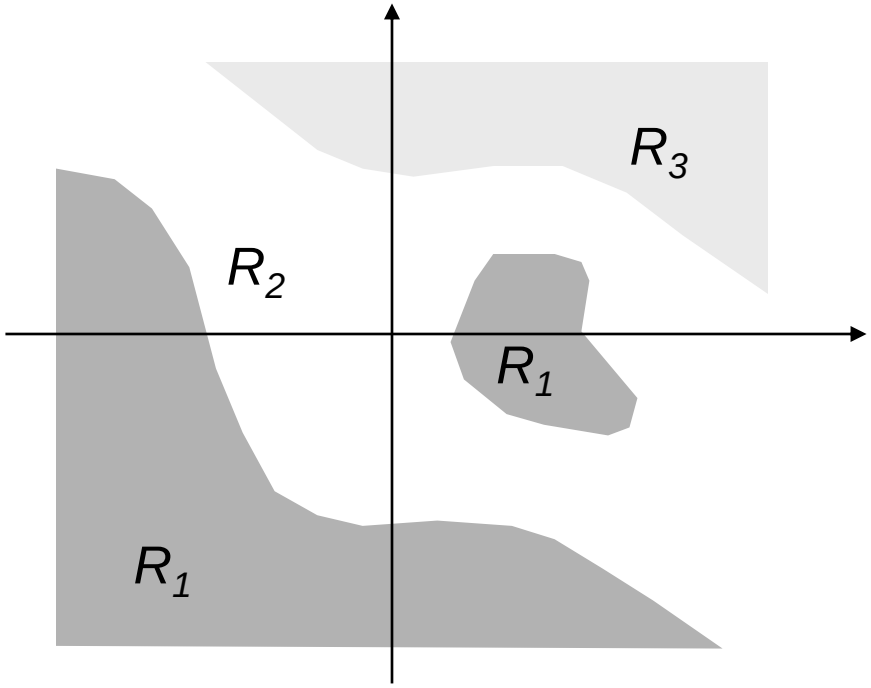


Decision is simple in this case!

Decision Regions



Decision Regions



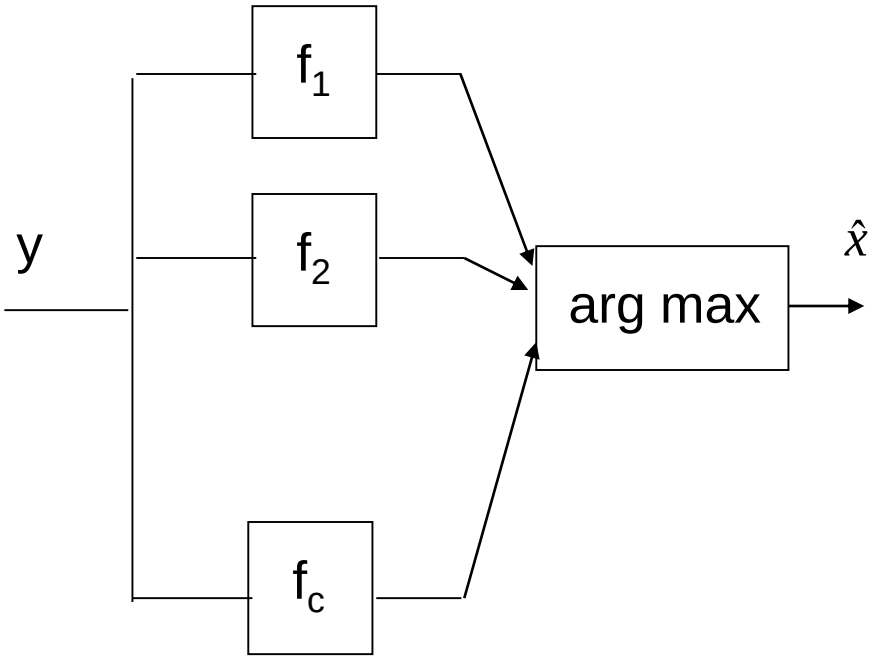
A classifier defines a partition of the feature space with c disjoint regions denoted by decision regions**:** R_1 , …, R_c.

The decision region R_i is the set of all the patterns classified in class i.

Discriminant functions

Functions $f_1, \ldots, f_c$, $(f_i: S \rightarrow R)$ are a set of discriminant functions for a classifier C if and only if

$$\hat{x} = \arg\max_i f_i(y)$$



Evaluation

A classifier can be evaluated using the confusion matrix $P=(P_{ij})$:

$$P_{ij} = P\{\hat{x} = j \mid x = i\} \qquad P_{ij} = \int_{R_j} p(y \mid i)dy$$

i.e. the probability of classifying an observation in class j if it was generated by class i.

Example:

$$P = \begin{bmatrix} 0.85 & 0.05 & 0.1 \\ 0 & 0.98 & 0.02 \\ 0.07 & 0.03 & 0.9 \end{bmatrix}$$

errors

erros

The most frequent error consists of classifying in class 3 patterns of class 1.

It is often difficult to compute the integral. The confusion matrix is often estimated applying the classifier to a set of data previously classified without errors (test set). The elements of the confusion matrix are approximated by the relative frequency of each type of error.

Error Probability

The probability of classification error is given by

$$P_e = 1 - \sum_{i=1}^{c} P_{ii} P_i$$

where

P_{ii} - diagonal element of the confusion matrix
P_i - *a priori* probability of class *i*

Example

$$P = \begin{bmatrix} 0.85 & 0.05 & 0.1 \\ 0 & 0.98 & 0.02 \\ 0.07 & 0.03 & 0.9 \end{bmatrix} \qquad \pi = \begin{bmatrix} .2 \\ .7 \\ .1 \end{bmatrix} \qquad P_e = 0.054$$

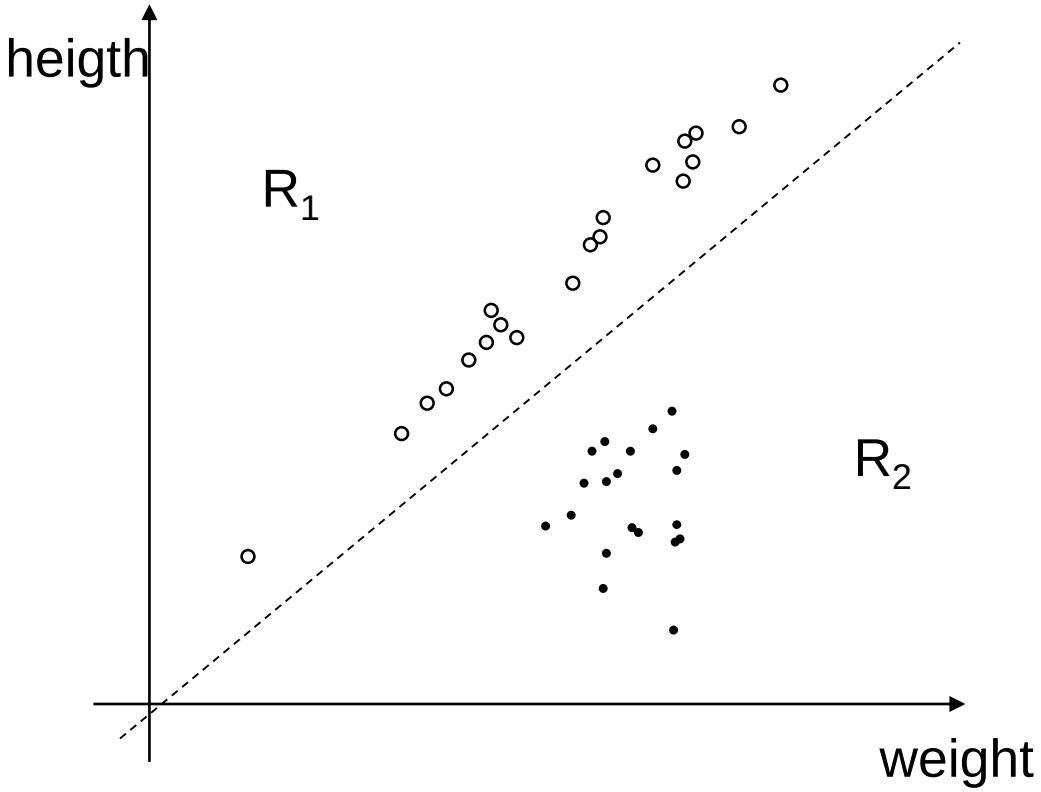
Design of a Classifier

A classifier is designed using the joint distribution of the data and classes or using examples of correct classifications (training set).

The estimation of a classifier from known decisions is denoted by *supervised training*.

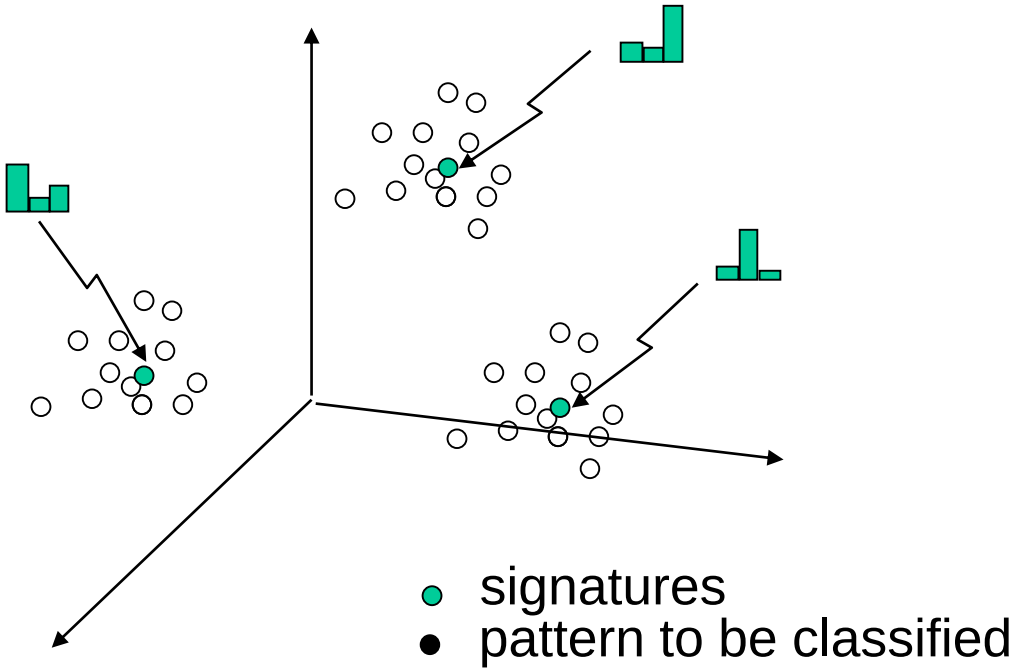
Visual Method

When the patterns belong to R^2 the classifier can be defined in a visual way



Method of the Nearest Signature

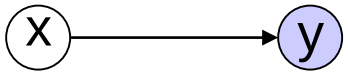
The method of the nearest signature consists of defining a prototype for each class (e.g., mean vector) and classifying the observations into the class of the nearest prototype.



The prototypes are often denoted as signatures.

Optimal Classifiers
(known distribution)

Inference and Classification



y – feature
x - classe

Problem: estimate x, given y ⟶ *inference problem*

x belongs to a finite set of classes: *W={1, .., c}*

The decision is based on the *a posteriori* distribution.

$$P(x \mid y) = kp(y \mid x)P(x) \qquad x \in \Omega$$

MAP Classifier

Classification law

$$\hat{x} = \arg\max_{x} p(y \mid x)P(x)$$

The discriminant functions are

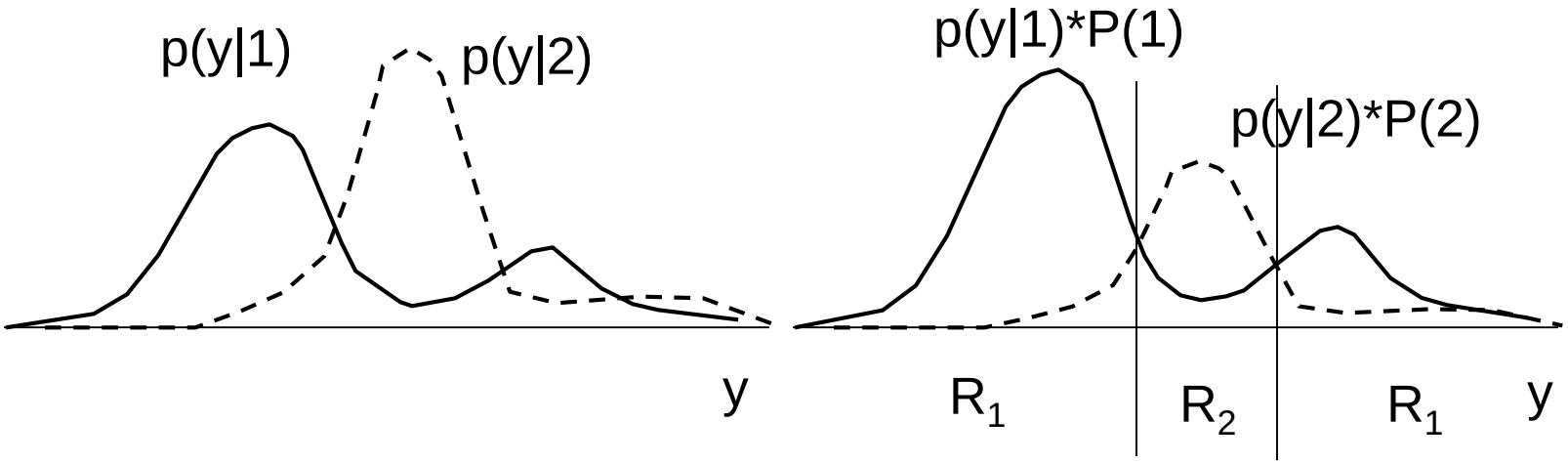
$$f_X(y) = p(y \mid x)P(x) \qquad x = 1,...,c$$

or

$$g_X(y) = \log p(y \mid x) + \log P(x) \qquad x = 1,...,c$$

A MAP classifier is optimal in the sense that it has the smallest probability of classification error.

Example



Example

Class and observation sets : $\Omega = \{1,2,3\}$ $y \in \{a,b,c,d\}$

Prior : $P(1) = 0.5, P(2) = 0.2, P(3) = 0.3$

a posteriori distribution

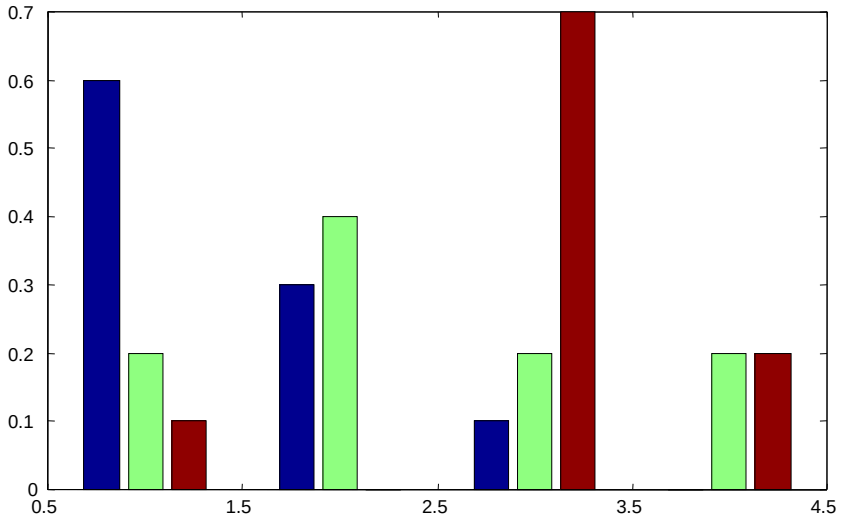
P(y|x)	a	b	c	d
1	0.6	0.3	0.1	0
2	0.2	0.4	0.2	0.2
3	0.1	0	0.7	0.2

P(x|y)	a	b	c	d
1	**0.81**	**0.65**	0.17	0
2	0.11	0.35	0.13	0.4
3	0.08	0	**0.7**	**0.6**

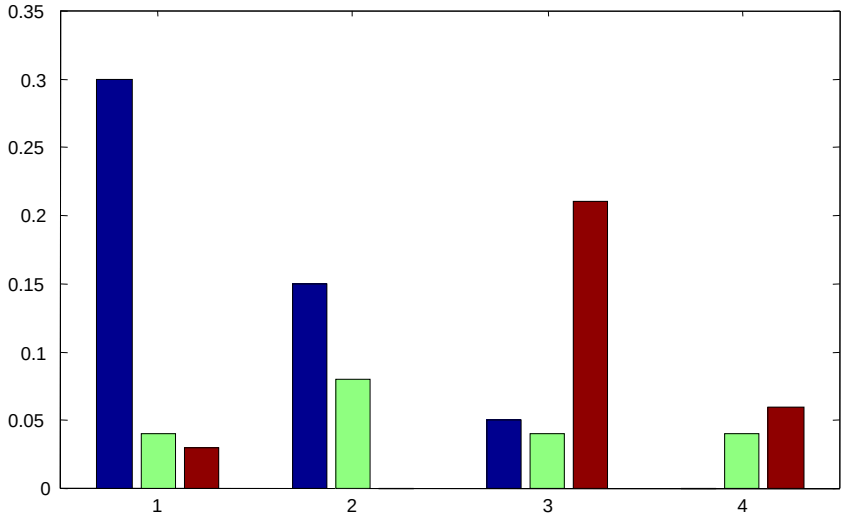
Decision regions: R_1={a,b} R_2=ϕ R_3={c,d}

Example (cont.)

Data model *p(y|x)*



*P(y|x)*p(x)*



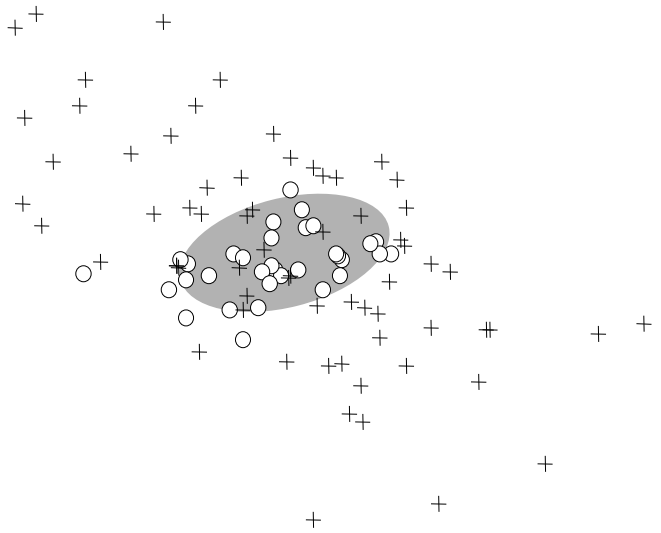
Example

Data:

P(1)=1/3 P(2)=2/3

$$p(y \mid i) = N(0, R_i)$$

$$R_1 = \begin{bmatrix} 2 & .2 \\ .2 & 1 \end{bmatrix} \quad R_2 = \begin{bmatrix} 10 & -6 \\ -6 & 10 \end{bmatrix}$$



Error percentage: 19%

Bayes Classifier

Decision law:

$$\hat{x} = \arg\max_i c_i(y) \qquad c_i(y) = \sum_{x=1}^{c} c(x,i)P(x \mid y)$$

$c_i(y)$ cost of classifying y in class i

The Bayes classifier minimizes the decision risk:

$$\Re = E\{c(x,\hat{x}) \mid y\} \qquad c(x,\hat{x}) \text{ cost}$$

The MAP classifier is a special case of the Bayes classifier when c(i,j)=1-d_{ij} i.e., when all the same cost is assigned to all errors.

Binary Classification

The Bayes classifier compares the likelihood ratio with a threshold

$$\frac{p(y/x=2)}{p(y/x=1)} \underset{1}{\overset{2}{\gtrless}} t \qquad t = \frac{(c_{12}-c_{11})P_1}{(c_{21}-c_{22})P_2}$$

The decision costs and the prior only influence the threshold.

Example

Sets of classes and observations : $\Omega = \{1,2,3\}$ $\quad y \in R$

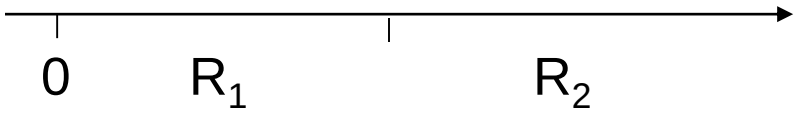
Prior : $P(i)$ $\quad$ observation model : $p(y/x) = \alpha_x e^{\alpha_x y}, y > 0$

Bayes Classifier:

$$y \underset{1}{\overset{2}{\gtrless}} t \qquad t = \frac{1}{\alpha_1 - \alpha_2} \log \frac{\alpha_1 (c_{12} - c_{11}) P_1}{\alpha_2 (c_{21} - c_{22}) P_2} \;, \quad \alpha_2 < \alpha_1$$

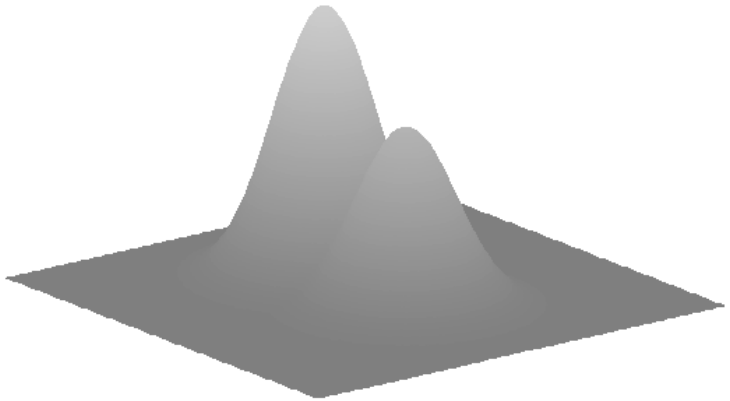
The threshold t depends on the cost matrix and a priori probabilities.

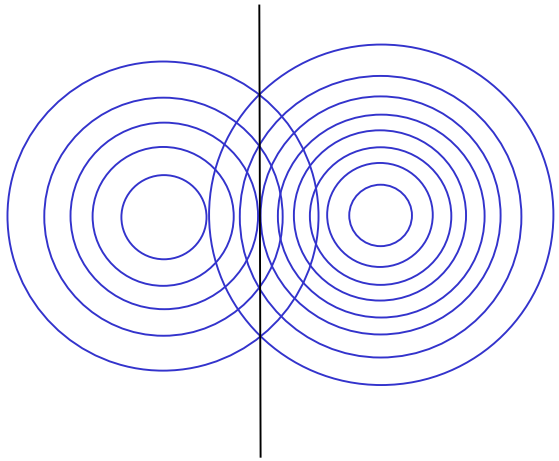
The Bayes decision is equivalent to a comparison with a threshold.



Gaussian Case

($R_i=s^2I$)





Discriminant function

$$f_i(y) = \| y - \bar{y}_i \|^2 + \log P_i$$

$\bar{y}_i$ is the signature of class i

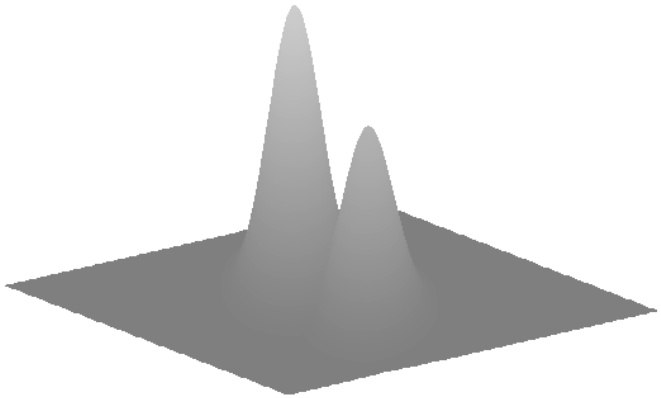
Linear discriminant function

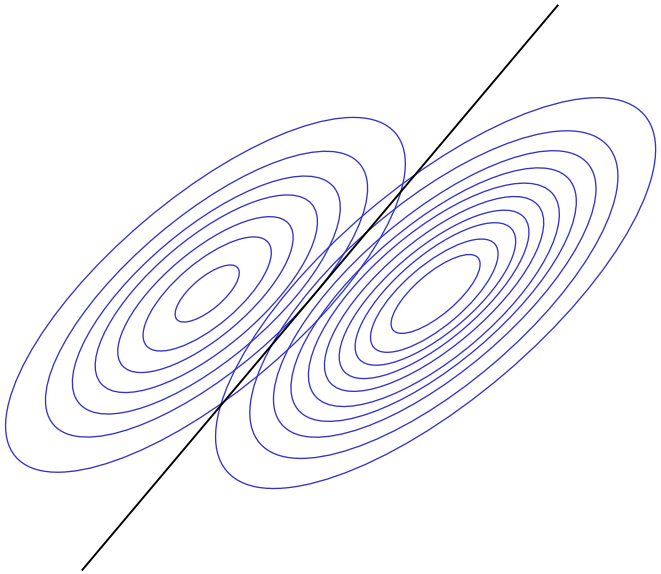
$$g_i(y) = \bar{y}_i' y + \bar{y}_i' \bar{y}_i + \log P_i$$

(classes equiprováveis: classificador de assinatura mais próxima)

Gaussian Case

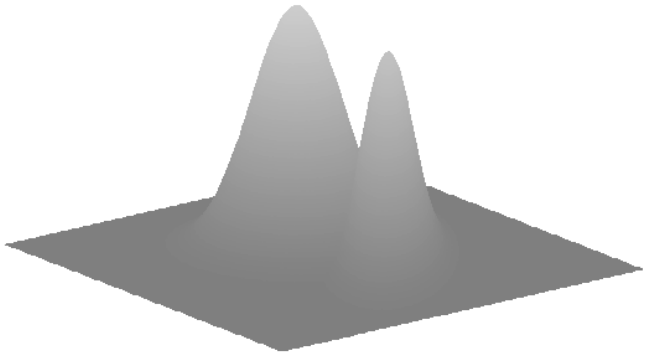
(R_i=R)

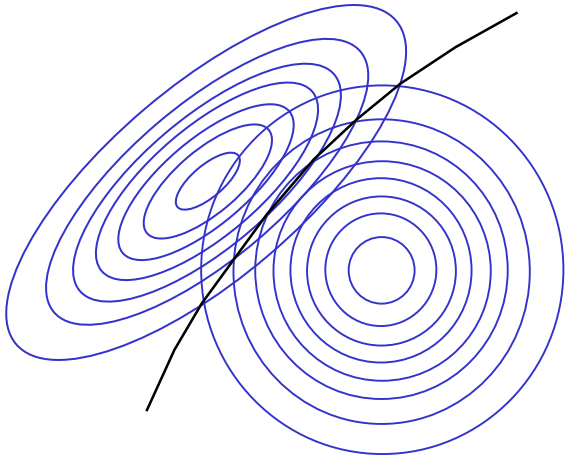




Gaussian Case

(different covariances)





Supervised Learning
(unknown distributions)

Supervised Learning

- estimation of the probabilistic model
- estimation of the discriminant functions

Training Set

The classifier is designed using a set of known feature vectors and decisions, denoted as **training set**.

The training set can be split into c subsets (one per class) containing the patterns associated with each class.

$$Y = \bigcup_{i=1}^{c} Y_i \qquad Y_i \text{ training patterns of i th class}$$

Estimation of the Probabilistic Model

MAP Classifier:

$$f_i = p(y/i)P(i)$$

$p(y/i)$ → relative frequency

$P(i)$ → probability density function

Both terms can be estimated using the training set Y.

Note: p(y/i) is estimated using the data associated to the i-th class

Estimation of P(y/Y)

Hypothesis: we know a parametric model, p(y/θ), with unknown θ.

How to estimate the density of y from Y ?

The Bayesian solution is

$$P(y/Y) = \int P(y/\theta)P(\theta/Y)d\theta$$

This integral can not be analytically computed in most cases.

An alternative is the use of numerical integration methods e.g., Monte Carlo. In practice a suboptimal approach is adopted :

$$\text{Hypothesis} : P(\theta/Y) = \delta(\theta - \hat{\theta}) \quad \text{therefore} \quad P(y/Y) = P(y/\hat{\theta})$$

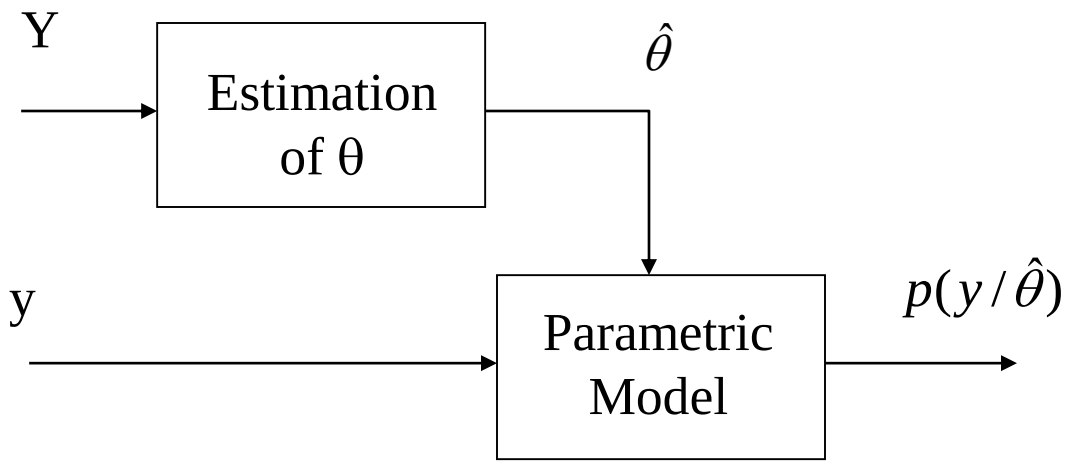
$\hat{\theta}$ is an estimate

Plug in Classifier

In practice, p(y/Y) is computed in two steps:

1. computate an estimate $\hat{\theta}$
2. replace $p(y/Y) = p(y/\hat{\theta})$

This procedure is sub-optimal

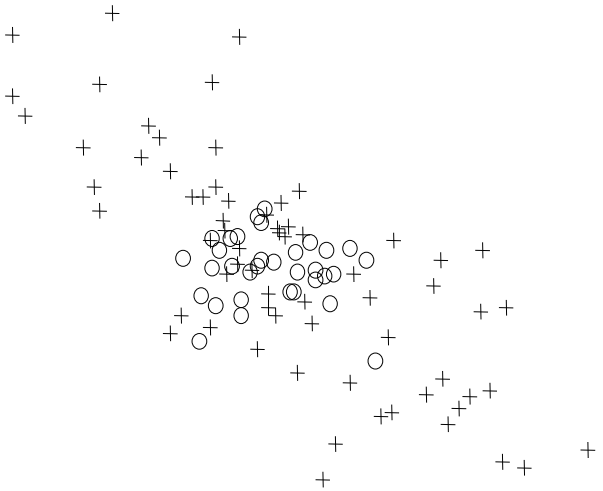


Estimation Methods

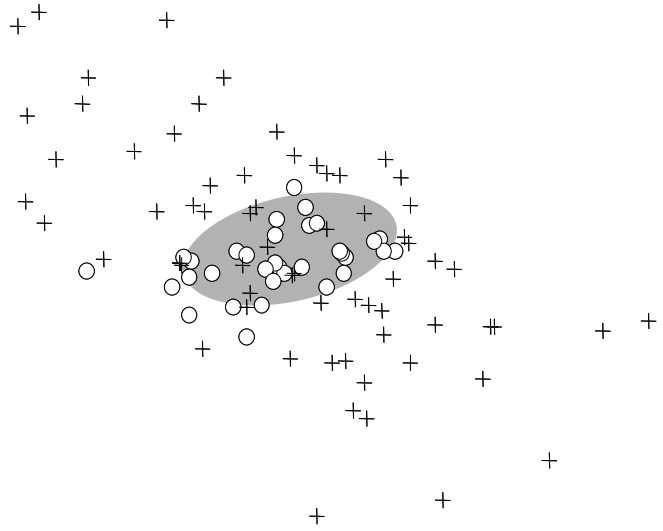
The methods adopted to estimate the unknown parameters are standard estimation methods e.g.,

- maximum likelihood
- minimum variance (minimum mean squared error)
- maximum a posteriori
- EM

Example



100 training patterns



100 test patterns

Confusion matrix:

24	9
12	55

Error percentage: 21%

Method of Parzen

This is a non parametric method. The density function is approximated by a sum of non-negative functions with unit integral, denoted as windows, centered at the training paterns. The intuitive idea is to distribute the "mass" of each training pattern by the neighboring points.

$$\hat{p}(y) = \sum_{\tilde{y} \in Y} w(y - \tilde{y})$$

w is denoted by window

Strong points: it is a general method since it does not assume any knowledge about the data distribution and easy to program.

Weak points: needs many training patterns (the number of patterns exponentially increases with the dimension of the feature space) and it is slow.

Adaptive Windows

The window *w* may be point dependent (adaptive)

$$\hat{p}(y) = \sum_{\tilde{y} \in Y} w_y(y - \tilde{y})$$

For example: narrow windows can be used in regions with a high density of points and large windows in regions with few data points.

A special case consists of choosing a binary window of finite support chosen in such a way that only k training patterns are inside the window suport:

$$w_y(y') = \begin{cases} 1 & \text{se } \| y - y' \| < \delta \\ 0 & \text{c.c.} \end{cases}$$

adaptative δ

Nearest Neighbor Classifier

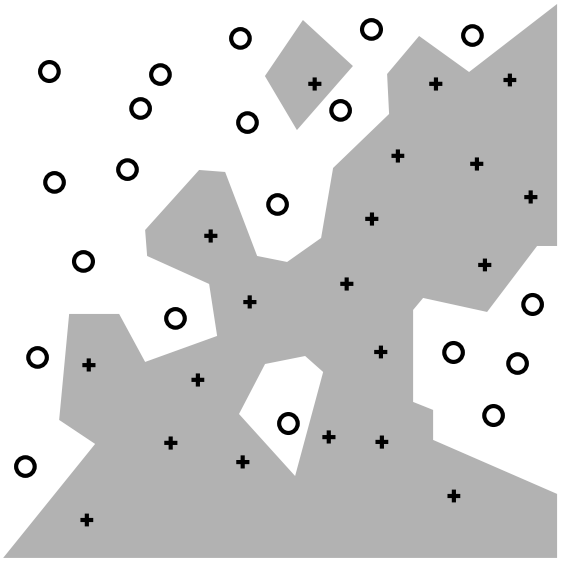
Let Y be a training set with classified patterns. The nearest neighbor classifier classifies each observation y in two steps:

determine the pattern $\tilde{y} \in Y$ nearest to y

classify y in the class of $\tilde{y}$

Example

The decision regions are Voronoi cells.

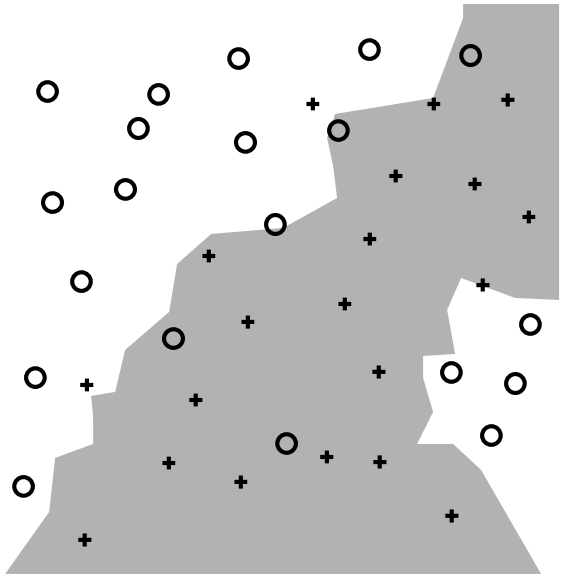


k-Nearest Neighbor Classifier

The nearest neighbor classifier determines the k training patterns closest to the pattern y to be classified; y is classified in the most voted class.

Example

k=3 neighbors



Asymptotic Properties

Is the nearest neighbor classifier a good classifier ?

Proposition (Cover, Hart, 1967)

Let Pe* be the probability of error of the MAP classifier in a classification problem with c classes. Then the probability of error of the nearest neighbor classifier converges in L_1 to a value

$$P_e \leq P_e^* (2 - \frac{k}{k-1} P_e^*) < 2P_e^*$$

when the number of training patterns tends to infinity.

Proof: see Duda e Hart, 1973 ou Ripley, 1996.

Complexity

The nearest neighbor classifier is one of the simplest classifiers. It is therefore is one of the first to be used when we have a data classification problem. However it is not optimal and has computational drawbacks.

The computation of the nearest neighbors is simple but slow. There are fast methods to perform this operation (Ripley, 1996) as well as pre-processing methods of the training set:

- outlier elimination
- data reduction (elimination of data points which do not influence the decision boundary)

Multiedit Algorithm

(Denvijver, Kittler, 1982)

The objective is the elimination of spurious points in the training set.

1. Define a current set with all traning patterns.
2. Separate the current set into V>2 disjoint subsets. Use pairs of subsets as traing and test sets.
3. For each pair classify the test patterns using the k-nn with the training patterns.
4. Eliminate from the current set all the test patterns which were incorrectly classified.
5. If some patterns were eliminated return to step 2.

Condensation

The condensation algorithms try to reduce the computational complexity of the nearest neighbor classifier reducing the number of training patterns without degradation of performance. Example:

Hart algorithm (1968)

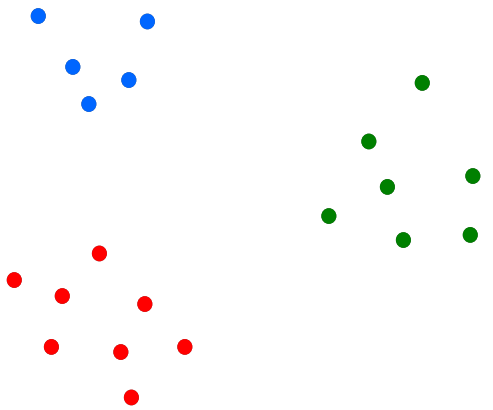
1. Initialize the training set with the first pattern a an auxiliary set (garbage) with the rest.
2. Classify each pattern of the auxiliary set with the nearest neighbor classifier. If the pattern was misclassified it is moved to the traning set.
3. Return to 2 until there is no more changes or the auxiliary set is empty.

Other issues

Several issues were not addressed:

- feature extraction and selection
- assessment of the classifier performance

Métodos de agrupamento



Objectivo: procurar grupos de pixels com características semelhantes

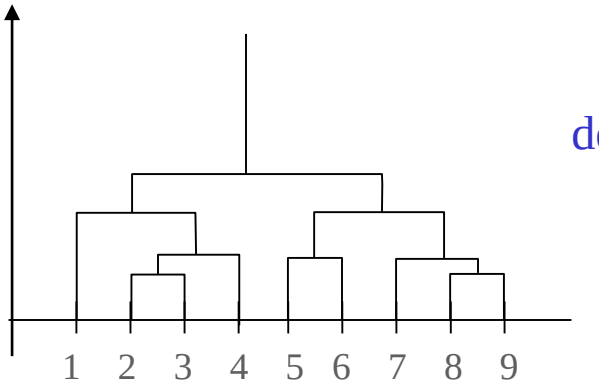
Métodos hierárquicos

Tem associada uma noção de escala.

métodos partitivos – dividem recursivamente o domínio da imagem em regiões cada vez mais pequenas.

métodos agomerativos – associam regiões elementares para formarem regiões maiores com base num critério de homogeneidade.

Métodos aglomerativos



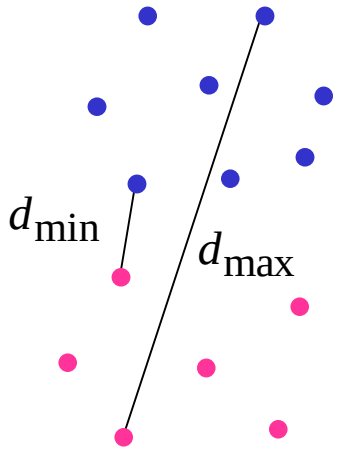
dendograma

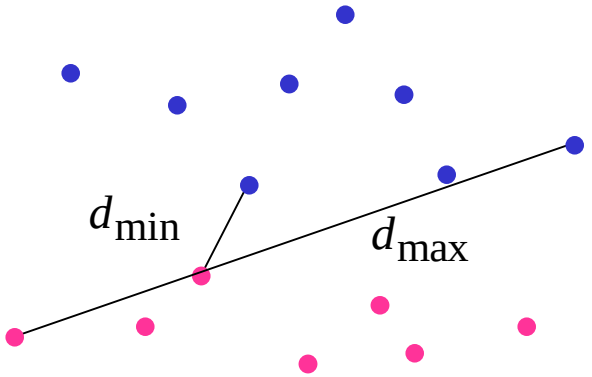
inicialização: definir uma colecção de regiões elementares (p.ex., pixels)

ciclo: até se obter um única região
- determinar o par de regiões mais próximas e fundi-los
- registar num dendograma as regiões associadas e a distância entre elas

segmentação: escolher uma distância máxima de fusão e obter os grupos de dados por análise do dendograma

Distância entre grupos

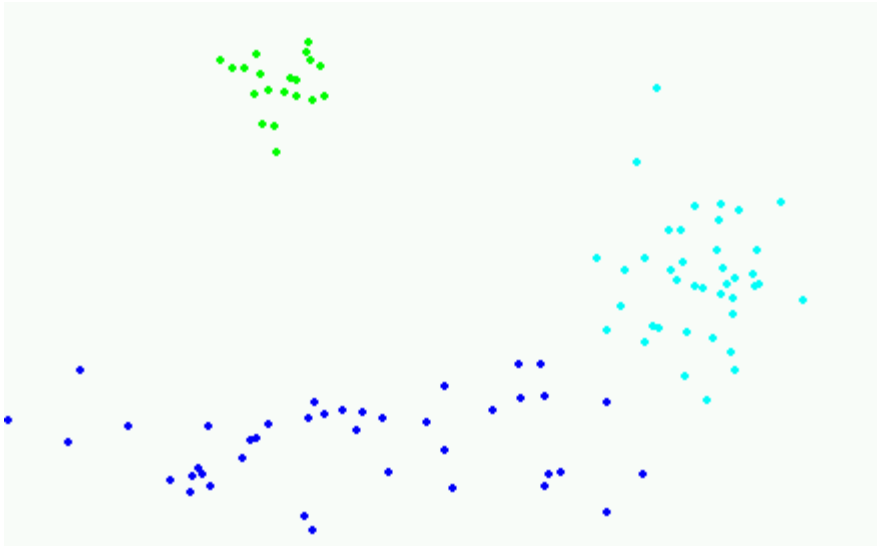


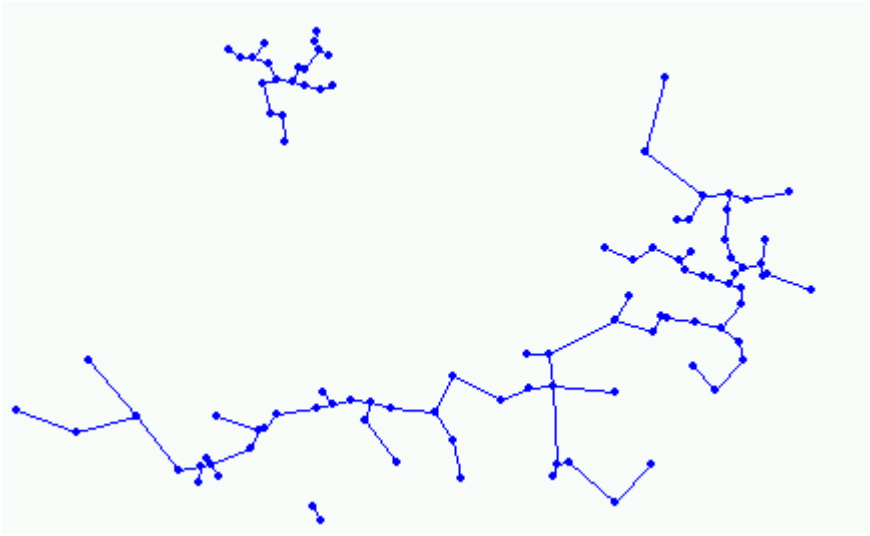


distância do mínimo $d_{\min} = \min_{x \in X, y \in Y} \| x - y \|$ método de ligação simples

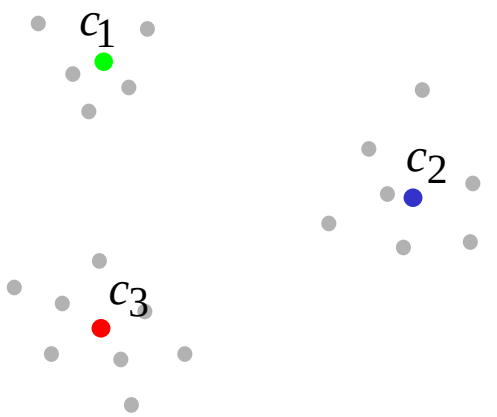
distância do máximo $d_{\max} = \max_{x \in X, y \in Y} \| x - y \|$ método de ligação completa

Exemplo





Método de k-Médias



Separar o conjunto de dados X em k subconjuntos disjuntos X_k

Aproximar os dados em cada subconjunto X_k por um centróide c_k

Critério

$$E = \sum_{i} \sum_{x \in X_i} \| x - c_i \|^2$$

problema de optimização

Método de k-Médias (2)

Inicialização: escolher valores para os k centróides (p.ex., k observações)

Ciclo

classificação: classificar os padrões na classe com centróides mais próximos

$$x \in X_i \quad \text{sse} \quad \| x - c_i \| < \| x - c_k \| \quad \forall k \neq i$$

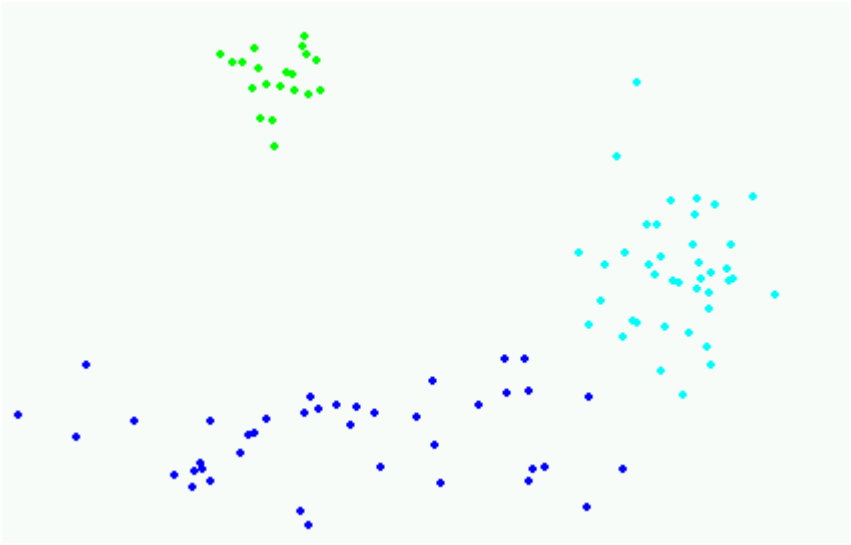
actualização: recalcular os centróides

$$c_k = \frac{1}{\#X_k} \sum_{x \in X_k} x$$

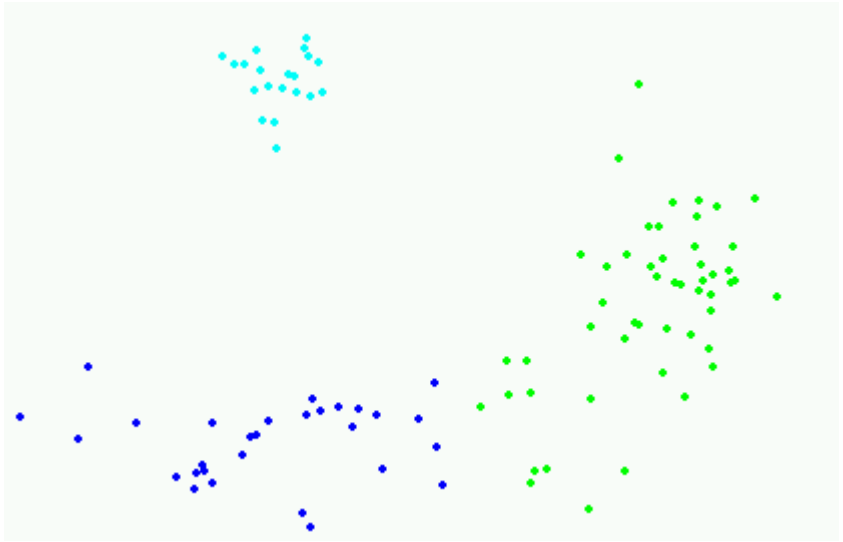
(se a distância for euclidiana)

Exemplo

dados



resultados do k-médias



Nota: mostra-se a classe correcta (cor) que não é usada pelo algoritmo de k-medias

Aplicação do K-médias

Segmentação com o método k-médias no espaço RGB, 9 classes

Exercises

Let y be an observation produced by a binomial distribution with parameter α_i which depends on the class i. Determine the decision law and decision regions of the MAP classifier, knowing that they occur with probabilities P(1), P(2).

Determine the decision regions of a MAP classifier knowing that the observation y is generated by each class according to normal distributions N(0,1), N(0,4) . The a priori distribution of the classes is given by P(1), P(2).

Let y_1, ..., y_n be independent realizations of a random variable characterized by the following conditional distributions:

$$p(y/x=1) = N(1,1) \qquad p(y/x=2) = \begin{cases} 2e^{-2y} & y>0 \\ 0 & c.c. \end{cases}$$

Determine the classification law and the decision regions of the MAP classifier.

Exercises

Determine the error probability of a MAP classifier knowing that y is generated by one of two classes with distribution

$$p(y/i) = \begin{cases} \alpha_i e^{-\alpha_i y} & y>0 \\ 0 & \text{c.c.} \end{cases} \qquad i = 1,2$$

The classes are equiprobable.

Determine a Bayes classifier for vector random variables with normal distributionknowing that the cost matrix is

$$C = \begin{bmatrix} 0 & 5 \\ 1 & 0 \end{bmatrix}$$

And we have a training set with realizations of y produced by both classes:

$$X_1 = \{(-2,4),(-5,3),(-3,5),(-2,6),(-4,3),(-3,4)\} \qquad X_2 = \{(2,3),(1,1),(3,2),(2,2)\}$$

Bibliografia

•Jorge S. Marques, Reconhecimento de Padrões Métodos Estatísticos e Neuronais, IST Press, 1999.

•B. Ripley, Pattern Recognition and Neural Networks, Cambridge University Press, 1996.