# Ciência de dados

KNN

## K-Nearest Neighbors

→ instances represented as points in the Euclidean space

→ given an unknown instance z to be classified, choose the k nearest neighbors and classify z according to the class of their majority

(k) Start with $k=1$; repeat with $k=k+2$
↓
choose the value of k for which the error rate is minimum

### More than k equally-distant neighbors

* they are all of the same class → instance is classified in the same class
* else: the classifier can't choose what neighbor to consider: IT DOES NOT CLASSIFY THE INSTANCE

### Ties (non-binary classification)

* If there is no majority class among neighbors → use k-1 nearest neighbors to determine the class
* If the tie persists → use k-2 nearest neighbors to determine the class

...

(Binary classification: k should be odd nr)

- Storing the data → no model construction
- constructs local approximations to the target function
- all computation takes place at classification time

→

## WHEN

- → instances map to points in $R^d$
- → less than 20 features → normalized
- → lots of training data

## ADVANTAGES

- → training is very fast
- → learn complex target functions
- → do not loose information

## DISADVANTAGES

- → slow at classification time
- → preparing and indexing training samples into search trees reduces time
- → easily fooled by irrelevant attributes

KNN creates a VORONOI diagram (composed by VORONOI CELLS)

↓

partition of a plane into regions based on distance to specific points (seeds) of the plane

KNN - dúvidas

→ K = 7 → 9 pontos (4 mais afastados) → 2N, 2P

e os outros não P

↓

pode-se classificar como P?

odd - ímpar

even - par

Exercise

3-NN ... (0,5) = a

d(

b = ... $\sqrt{(\ldots,25)^2 + (0,5-0,75)^2} =$

... $\sqrt{(\ldots,25)^2 + (-0,25)^2} \simeq 0,35$

$c = d(a, (1,25; 0,25)) = \sqrt{(1-1,25)^2 + (0,5-0,25)^2} \simeq 0,35$

$d = d(a, (0,75; 0,75)) = \sqrt{(1-0,75)^2 + (0,5-0,75)^2} \simeq 0,35$

$e = d(a, (0,75; 0,25)) = \sqrt{(1-0,75)^2 + (0,5-0,25)^2} \simeq 0,35$

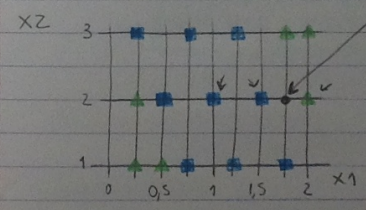
Os ponto com $x \pm 0,25$ e $y \pm 0,25$ → 4 pontos

Como esses 4 pontos não pertencem todos à mesma classe, 3-NN não conseguiu classificar.

(c)

Exercise 2

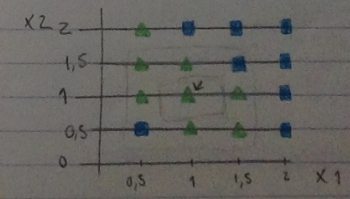
K = 3 instance (1,75; 2)




(c) It is not able to classify

NEG POS

Exercise 3




X1 and X2 numerical

Y binary: P and N

LEAVE-ONE-OUT STRATEGY

(c)/(d)

K = 5??

## Exercise 4

a) 3 binary attributes

$f = \overline{x2} \cdot x1 \cdot x3 \vee \overline{x1} \cdot x2 \cdot x3$

| A | B | C | Class | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | (1,0,0) v 100 |
| 0 | 0 | 1 | 0 | 011 v 101 |
| 0 | 1 | 0 | 0 | 000 v 110 |
| 0 | 1 | 1 | 1 | 001 v 111 |
| 1 | 0 | 0 | 0 | 110 v 000 |
| 1 | 0 | 1 | 1 | 111 v 001 |
| 1 | 1 | 0 | 0 | 010 v 010 |
| 1 | 1 | 1 | 1 0 | 011 v 011 |

↓

011

b) Accuracy?

$$\frac{TP + TN}{All}$$

$$P(Y|X) = \frac{P(X|Y)\,P(Y)}{P(X)}$$

"likelihood" "prior"

↳ normalizer

## Naive Bayes

A and B independent $\Leftrightarrow$ $P(A \cap B) = P(A) \times P(B)$

- $P(A \cap B) = P(A|B) \times P(B)$
- $P(A \cap B) = P(B|A) \times P(A)$

Naive Bayes

→ estimates the posterior probability for each class $c_i$, given $\vec{x}$ and chooses the most probable class

$$\hat{y} = \arg\max_{c_i} \{P(c_i) \times P(\vec{x}|c_i)\}$$

PRIOR PROBABILITY (probability before seeing any data)

LIKELIHOOD (probability of observing $\vec{x}$, given that it belongs to $c_i$)

$P(c_i|\vec{x})$: POST PROBABILITY (probability for each class given the observed instance $x$)

$P(\vec{x})$: probability for the instance to be observed, fixed for each instance

↓

Given $\vec{x} = (A_1 = v_{x1}, \dots, A_d = v_{xd}) \rightarrow$

$$\hat{y} = \arg\max_{c_i} \{P(c_i) \times P(\vec{x}|c_i)\}$$

$$\boxed{\hat{y} = \arg\max_{c_i} \{P(c_i) \times P(A_1 = v_{x1} \wedge \dots A_d = v_{xd}|c_i)\}}$$

PRIOR PROBABILITY: $\hat{P}(c_i) = \frac{n_i}{n}$

LIKELIHOOD

- numeric attributes: $\hat{P}(x_j|c_i) = f(x_j|\mu_{ij}, \sigma^2_{ij})$

$$\hat{\mu}_{ij} = \frac{1}{n_i} \sum_{x_k \in A_j} x_{ik} \qquad \sigma^2_{ij} = \frac{1}{n_i} Z^T_{ij} Z_{ij}$$

$$Z_{ij} = A_j - 1 \cdot \hat{\mu}^T_{ij}$$

- symbolic attributes: $\hat{P}(x_j|c_i) = \frac{|\{\vec{z} \in D_i : \vec{z} \cdot A_j = x_j\}| + 1}{n_i + m_j}$

$m_j = |dom(A_j)|$

ADVANTAGES:

→ easy to implement
→ greatly reduces the computation cost
→ good results obtained in most of the cases

WHEN TO USE:

→ moderate or large training set available
→ attributes that describe instances are conditionally independent given classification

MEAN: $\mu = \frac{1}{n} \sum_{i=1}^{n} x_i$

STANDARD DEVIATION (SAMPLE): $\sigma = \left[ \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \mu)^2 \right]^{0.5}$

↳ a (n) é POPULATION

NORMAL DISTRIBUTION: $f(x) = \frac{1}{\sqrt{2\pi}\,\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$

↓

Para atributos numéricos

Naive Bayes DUVIDAS:

→ fórmula σ (n-1 ou n) ← sample
no dominador) ↓ population

→ quando temos nºs (numerical) e a coordenada não binários (1 ou 0) fazemos normal ou com a função de densidade?

Exercise 1

attributes A and B binary
class x binary

a) instance (A,B)

$P(C=P \mid (A,B)) = P(C=P) \times P($

$P(C=N \mid (A,B)) = P(C=N) \times P(A \mid C=N) \times P(B \mid C=N)$

$P(X=P \mid (A,B)) = \frac{3}{8} \times 1 \times \frac{1}{3} = \frac{1}{8}$

$P(X=N \mid (A,B)) = \frac{5}{8} \times \frac{2}{5} \times \frac{4}{5} = \frac{1}{5} > \frac{1}{8}$

(B)

b) $P(AB \mid X) = 1/3$ $P(A\bar{B} \mid X) = 2/3$ $P(\bar{A}B \mid X) = 0$
$P(\bar{A}\bar{B} \mid X) = 0$

Exercise 2

instance (T,T,F) $P(P) = \frac{2}{8} = \frac{1}{4}$ $P(N) = \frac{6}{8} = \frac{3}{4}$

$P(X=P \mid (T,T,F)) = P(X=P) \times P(A=T \mid X=P) \times P(B=T \mid X=P) \times P(C=F \mid X=P)$

$= \frac{1}{4} \times 1 \times 0 \times \frac{1}{2} = 0$

$P(X=N \mid (T,T,F)) = P(X=N) \times P(A=T \mid X=N) \times P(B=T \mid X=N) \times P(C=F \mid X=N)$

$= \frac{3}{4} \times \frac{3}{6} \times \frac{3}{6} \times \frac{4}{6} =$

$= \frac{\cancel{3}}{4} \times \frac{1}{2} \times \frac{1}{\cancel{2}} \times \frac{\cancel{2}}{\cancel{3}} = \frac{1}{8} > 0$

↓

NEGATIVE (B)

está mal → não se calcula função de densidade p/ atributos ordinais

Exercise 3 $n = 15$ instances $P(P) = \frac{6}{15}$ $P(N) = \frac{9}{15}$

Instance (1.0, 0.5)

FOR POSITIVES

$\boxed{A:}$

$\mu_A = 1{,}708$

$$\sigma_A = \left(\frac{1}{6}(0{,}501264 + 0{,}209764 + 0{,}043264 + 0{,}001764 + 0{,}293764 + 0{,}627264)\right)^{0{,}5}$$

$$= \left(\frac{1{,}677084}{6}\right)^{0{,}5} \simeq 0{,}529$$

$\boxed{B:}$

$\mu_B = 0{,}625$

$$\sigma_B = \left(\frac{1}{6}\cdot(0{,}015625 \times 6)\right)^{0{,}5} = 0{,}125$$

FOR NEGATIVES:

$\boxed{A:}$

$\mu_A = 1{,}306$

$$\sigma_A = \left(\frac{1}{9}\cdot(1{,}115136 + 0{,}649636 + 0{,}309136 \times 2 + 0{,}003136 + 0{,}197136 + 0{,}481636 + 0{,}891136 \times 2)\right)^{0{,}5}$$

$$= \left(\frac{4{,}847224}{9}\right)^{0{,}5} \simeq 0{,}734$$

$\mu_B = 0{,}417$

$$\sigma_B = \left(\frac{1}{9}\cdot(0{,}027889 + 0{,}006889 + 0{,}027889 + 0{,}110889 + 0{,}027889 \times 2 + 0{,}006889 + 0{,}110889 + 0{,}027889)\right)^{0{,}5}$$

$$= \left(\frac{0{,}375001}{9}\right)^{0{,}5} \simeq 0{,}204$$

$$P(\text{Class} = P \mid A = 1.0 \wedge B = 0.5) = P(C=P) \times P(A=1.0 \mid C=P) \times P(B=0.5 \mid C=P)$$

$$= \frac{6}{15} \times 0{,}307957 \times 1{,}93 = 0{,}23774 \gg$$

$$P(A=1.0 \mid C=P) = f(1.0) = \frac{1}{\sqrt{2\pi}\cdot 0{,}529} \cdot e^{-\frac{(1-1{,}708)^2}{2 \times 0{,}529^2}} = \frac{1}{\sqrt{2\pi}\cdot 0{,}529} \cdot e^{\frac{-0{,}501264}{0{,}559682}} = 0{,}307957$$

$$P(B=0.5 \mid C=P) = f(0{,}5) = \frac{1}{\sqrt{2\pi}\cdot 0{,}125} \cdot e^{-\frac{(0{,}5-0{,}625)^2}{2 \times 0{,}125^2}} = \frac{1}{\sqrt{2\pi}\cdot 0{,}125} e^{-\frac{0{,}015625}{0{,}03125}} = 1{,}93$$

$$P(\text{Class} = N \mid A = 1.0 \wedge B = 0{,}5) = P(C=N) \times P(A=1.0 \mid C=N) \times P(B=0{,}5 \mid C=N) =$$

$$= \frac{9}{15} \times 0{,}32966 \times 0{,}032944 = 0{,}0065$$

$$P(A=1.0 \mid C=N) = f(1.0) = \frac{1}{\sqrt{2\pi}\cdot 0{,}734} \cdot e^{-\frac{(1-1{,}306)^2}{2 \times 0{,}734^2}} = \frac{1}{0{,}734\sqrt{2\pi}} \cdot e^{\frac{-0{,}093636}{1{,}077512}} = 0{,}32966$$

$$P(B=0{,}5 \mid C=N) = f(0{,}5) = \frac{1}{\sqrt{2\pi}\cdot 0{,}204} \cdot e^{-\frac{(\overset{0{,}5}{\cancel{x}} - 0{,}417)^2}{2 \times 0{,}204^2}} = \frac{1}{0{,}204 \times \sqrt{2\pi}} \cdot e^{\frac{-0{,}339889}{0{,}083232}} = 0{,}032944$$

(A) positive

## Exercise 4

12 instances $P(N) = \frac{5}{12}$ $P(P) = \frac{7}{12}$

X1 numerical X2 ordinal

| X1 | 0,5 | 0,5 | 0,5 | 1 | 1 | 1 | 1,5 | 1,5 | 1,5 | 2 | 2 | 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| X2 | 0,5 | 1 | 1,5 | 0,5 | 1 | 1,5 | 0,5 | 1 | 1,5 | 0,5 | 1 | 1,5 |
| C | P | P | N | P | P | N | N | N | P | P | N | P |

Classify instance (0,75; 1)

FOR POSITIVES:

$\mu_{x1} = 1,214$

$$\sigma_{x1} = \left(\frac{1}{7}(0,509796 \times 2 + 0,045796 \times 2 + 0,081796 + 0,617796 \times 2)\right)^{0,5} = 0,589$$

FOR NEGATIVES:

$\mu_{x1} = 1,3$

$$\sigma_{x1} = \left(\frac{1}{5}(0,64 + 0,09 + 0,04 \times 2 + 0,49)\right)^{0,5} = 0,5099$$

$$P(C=P \mid x1=0,75 \wedge x2=1) = P(C=P) \times P(x1=0,75 \mid C=P) \times P(x2=1 \mid C=P) =$$

$$= \frac{7}{12} \times 0,49663 \times \frac{2}{7} = 0,08277 >$$

$$P(x1=0,75 \mid C=P) = f(0,75) = \frac{1}{0,589 \times \sqrt{2\pi}} \cdot e^{\frac{-(0,75-1,214)^2}{2 \times 0,589^2}} =$$

$$= \frac{1}{0,589 \times \sqrt{2\pi}} \cdot e^{-\frac{0,215296}{0,693842}} =$$

$$= 0,49663$$

$$P(C=N \mid x1=0,75 \wedge x2=1) = P(C=N) \times P(x1=0,75 \mid C=N) \times P(x2=1 \mid C=N) =$$

$$= \frac{8}{12} \times 0,4373 \times \frac{2}{8} = 0,07288$$

$$P(x1=0,75 \mid C=N) = f(0,75) = \frac{1}{0,5099 \times \sqrt{2\pi}} \cdot e^{\frac{-(0,75-1,3)^2}{2 \times 0,5099^2}} = 0,4373$$

Ⓐ POSITIVE

## Exercise 5

15 instances $P(P) = \frac{2}{5}$ $P(N) = \frac{3}{5}$

X1 numerical, X2 ordinal

| X1 | 0,25 | 0,25 | 0,25 | 0,5 | 0,5 | 0,75 | 0,75 | 1 | 1,25 | 1,25 | 1,5 | 1,75 | 1,75 | 2 | 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| X2 | 1 | 2 | 3 | 1 | 2 | 1 | 3 | 2 | 1 | 3 | 2 | 1 | 3 | 2 | 3 |
| C | P | P | N | P | N | N | N | N | N | N | N | N | P | P | P |

Classify instance (1,75; 2)

FOR POSITIVES:

$\mu_{x1} = 1,125$

$$\sigma_{x1} = \left(\frac{1}{6}(0,765625 \times 2 + 0,390625 + 0,390625 + 0,765625 \times 2)\right)^{0,5} = 0,8$$

FOR NEGATIVES

$\mu_{x1} = 1$

$$\sigma_{x1} = \left(\frac{1}{9}(0,5625 + 0,25 + 0,0625 \times 2 + 0 + 0,0625 \times 2 + 0,25 + 0,5625)\right)^{0,5} = 0,4564$$

$$P(C=P \mid x1=1,75 \wedge x2=2) = P(C=P) \times P(x1=1,75 \mid C=P) \times P(x2=2 \mid C=P) =$$

$$= \frac{2}{5} \times 0,4426 \times \frac{2}{6} = 0,059018$$

$$P(x1=1,75 \mid C=P) = f(1,75) = \frac{1}{0,8\sqrt{2\pi}} \cdot e^{-\frac{(1,75-1,125)^2}{2 \times 0,8^2}} = 0,4426$$

$$P(C=N \mid x1=1,75 \wedge x2=2) = P(C=N) \times P(x1=1,75 \mid C=N) \times P(x2=2 \mid C=N) =$$

$$= \frac{3}{5} \times 0,22556 \times \frac{1}{3} = 0,0453$$

$$P(x1=1,75 \mid C=N) = \frac{1}{0,4564\sqrt{2\pi}} \cdot e^{\frac{-(1,75-1)^2}{2 \times 0,4564^2}} = 0,226556$$

Ⓐ POSITIVE

## Exercise 6

16 instances $\quad P(P) = \frac{8}{16} = \frac{1}{2} \quad P(N) = \frac{8}{16} = \frac{1}{2}$

X1 numerical, X2 numerical

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| X1 | 0,5 | 0,5 | 0,5 | 0,5 | 1 | 1 | 1 | 1 | 1,5 | 1,5 | 1,5 | 1,5 | 2 | 2 | 2 | 2 |
| X2 | 0,5 | 1 | 1,5 | 2 | 0,5 | 1 | 1,5 | 2 | 0,5 | 1 | 1,5 | 2 | 0,5 | 1 | 1,5 | 2 |
| C | N | P | P | P | P | P | P | N | P | P | N | N | N | N | N | N |

Classify instance (1,1)

FOR POSITIVES

(X1)

$\mu_{x1} = 0{,}9375$

$\sigma_{x1} = \left(\frac{1}{8}(0{,}1914 \times 3 + 0{,}0039 \times 3 + 0{,}3164 \times 2)\right)^{0,5} = 0{,}39$

(X2)

$\mu_{x2} = 1{,}125$

$\sigma_{x2} = \left(\frac{1}{8}(0{,}015625 \times 3 + 0{,}140625 \times 2 + 0{,}765625 + 0{,}390625 \times 2)\right)^{0,5} = 0{,}4841$

FOR NEGATIVES

(X1)

$\mu_{x1} = 1{,}5625$

$\sigma_{x1} = \left(\frac{1}{8}(1{,}1289 + 0{,}3164 + 0{,}0039 \times 2 + 0{,}1914 \times 4)\right)^{0,5} = 0{,}5266$

(X2)

$\mu_{x2} = 1{,}375$

$\sigma_{x2} = \left(\frac{1}{8}(0{,}765625 \times 2 + 0{,}390625 \times 3 + 0{,}015625 \times 2 + 0{,}140625)\right)^{0,5} = 0{,}5994789$

$P(C=P \mid X_1=1 \wedge X_2=1) = P(C=P) \times P(X_1=1 \mid C=P) \times P(X_2$

$= \frac{1}{2} \times 1{,}009877411 \times 0{,}79707 =$

$P(X_1=1 \mid C=P) = \frac{1}{\sqrt{2\pi} \times 0{,}39} \cdot e^{-\frac{(1-0{,}9375)^2}{2 \times 0{,}39^2}} = 1{,}00987$

$P(X_2=1 \mid C=P) = \frac{1}{\sqrt{2\pi} \times 0{,}4841} e^{-\frac{(1-1{,}125)^2}{2 \times 0{,}4841^2}} = 0{,}79707$

f(x) pode ser > 1 ??

$P(C=N \mid X_1=1 \wedge X_2=1) = P(C=N) \times P(X_1=1 \mid C=N) \times P(X_2=1 \mid C=N) =$

$= \frac{1}{2} \times 0{,}4282 \times 0{,}5472 = 0{,}117156$

$P(X_1=1 \mid C=N) = \frac{1}{\sqrt{2\pi} \times 0{,}5266} \cdot e^{-\frac{(1-1{,}5625)^2}{2 \times 0{,}5266^2}} = 0{,}4282$

$P(X_2=1 \mid C=N) = \frac{1}{\sqrt{2\pi} \times 0{,}5995} \cdot e^{-\frac{(1-1{,}375)^2}{2 \times 0{,}5995^2}} = 0{,}5472$

(A) POSITIVE

## Exercise 8

(C) FOR POSITIVES: $\mu = 0{,}438 \quad \sigma = 0{,}222$

FOR NEGATIVES: $\mu = 0{,}625 \quad \sigma = 0{,}144$

Classify Instance (1,1,0.75)

$P(P) = \frac{8}{12} = \frac{2}{3} \qquad P(N) = \frac{4}{12} = \frac{1}{3}$

$P(C=P \mid A=1 \wedge B=1 \wedge C=0{,}75) = P(C=P) \times P(A=1 \mid C=P) \times P(B=1 \mid C=P) \times P(C=0{,}75 \mid C=P)$

$= \frac{\cancel{2}}{3} \times \frac{\cancel{4}}{\cancel{8}} \times \frac{4}{8} \times 0{,}66935 = 0{,}111558$

| A | B | C | Class |
|---|---|---|---|
| 0 | 0 | 0,25 | N ? |
| 0 | 1 | 0,25 | P |
| 1 | 0 | 0,25 | P |
| 1 | 1 | 0,25 | N ? |
| 0 | 0 | 0,5 | N ? |
| 0 | 1 | 0,5 | P |
| 1 | 0 | 0,5 | P |
| 1 | 1 | 0,5 | N ? |
| 0 | 0 | 0,75 | N ? |
| 0 | 1 | 0,75 | P |
| 1 | 0 | 0,75 | P |
| 1 | 1 | 0,75 | N ? |

$P(C=0{,}75 \mid C=P) = \frac{1}{\sqrt{2\pi} \times 0{,}222} \cdot e^{-\frac{(0{,}75-0{,}438)^2}{2 \times 0{,}222^2}} = 0{,}66935$

$P(C=0{,}75 \mid C=N) = \frac{1}{\sqrt{2\pi} \times 0{,}144} e^{-\frac{(0{,}75-0{,}625)^2}{2 \times 0{,}144^2}} = 1{,}900739113$

$P(C=N \mid A=1 \wedge B=1 \wedge C=0{,}75) = \frac{1}{8} \times \quad \times 1{,}900 \quad =$

> (B) NEGATIVE

Exercise 9

A, B and C numeric attributes with values 0,25, 0.5 and 0,75

## Decision Trees

TREE
- Nodes : test for the value of an attribute
- +
- Branches : results of those tests
- +
- Leaves : classes

### ID3

Decision Trees | ID3 | C4.5 | CART

CONSTRUCTION : TOP-DOWN recursive divide-and-conquer manner

ALGORITHM
- training examples are at the root
- attributes are categorical
- examples are partitioned recursively based on selected attributes
- test attributes are selected on the basis of a heuristic or statistical measure

CONDITIONS FOR STOPPING PARTITIONING
- all samples for a given node belong to the same class
- no remaining attributes for further partitioning – MAJORITY VOTING (to classify the leaf)
- there are no samples left.

LEARNING / TRAINING :
- select the best attribute to test in the root node
- create a branch for each test result ; each branch will cover a subset of the training set
- for each subset, return to step 1

PRUNING : to identify and remove branches that are too tied to the training data (OVERFITTING)

METRICS : (CHOOSE THE BEST ATTRIBUTE)

1) INFORMATION GAIN

encoding information that would be gai

$$G(S,A) = E(S) - \sum_{v \in Dom(A)} \frac{\#\{x \in S : x.A = v\}}{\#\{x \in S\}} \times E$$

$$E(S) = \sum_{i=1}^{k} -\frac{\#\{x \in C_i\}}{\#\{x \in D\}} \times \log_2 \frac{\#\{x \in C_i\}}{\#\{x \in D\}} \leq$$

→ expected information needed to classify

node is pure → return

impurity is maximal → return

intermediate situation → apply measure in order to find information gain

→ amount of information, needed to decide if an arbitrary example in S belongs to P or N

$$E(p,n) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$$

class P class N

→ it grows each branch just deeply enough to perfectly classify all the training examples
→ it doesn't deal with numerical attributes neither with missing values
→ I.G. prefer attributes with more possible values

2) SPLIT INFO
3) Gini Index

## C4.5

METRIC (CHOOSING TESTS)
- GAIN RATIO (measure that balances the information gain with the number of possible values).

$$GainRatio(S,A) = \frac{G(S,A)}{splitInfo(S,A)}$$

↳ entropy of the set S ordered by the values of attribute A

$$splitInfo(S,A) = -\sum_{v_i \in Dom(A)} \frac{|\{x \in S : x.A = v_i\}|}{|\{x \in S\}|}\log_2\frac{|\{x \in S : x.A = v_i\}|}{|\{x \in S\}|}$$

→ Creates a BOOLEAN attribute to test NUMERIC attributes

Attribute A less than or equal to θ
↓
CHOOSE the TRESHOLD value θ
- order the training set according to values of A
- If A has m possibles values then there are m-1 interesting values for θ
  $Dom A = \{v_1, v_2, \dots, v_m\}$ [$v_n, v_{n+1}$[ same behaviour
  TEST all values $\forall n\ (1 \le n \le m-1)$ → choose the best one

## NOISY DATA
↳ random error or variance in a
↳ outliers
↓ methods
1) BINNING METHOD (first sort
equi-depth bins, then one
2) CLUSTERING (detect and remove outliers)
3) COMPUTER + HUMAN INSPECTION
4) REGRESSION (smooth by fitting the data into regression functions)

COMPARISON
- INFO GAIN: biased towards multivalued attributes
- GAIN RATIO: tends to prefer unbalanced splits in which one partition is much smaller than the other
- GINI INDEX: (=I.G + tends to favor tests that result in equal-sized partitions and purity in both partitions

## Missing values
- use training example anyway, sort through tree
  - If node n tests A, assign most common value of A among other examples sorted to node n
  - Assign most common value of A among other examples with same target value
  - Assign probability $p_i$ to each possible value of A

## CART
→ generate only binary trees
→ use of Gini Index and test the same attribute several times
→ prune with different choices : validation set
→ deal with missing values
→ return a forest
→ deal with CATEGORICAL + NUMERIC attributes
→ " " UNBALANCED DATA (automatically)

## Gini Index

$$gini(D) = 1 - \sum_{j=1}^{n} p_j^2$$

n: classes
↳ probability of class j in D
↳ purity of some set D

$$gini_{split}(D) = \sum_{i=1}^{m} \frac{N_i}{N} gini(D_i)$$

nº segments — $i^{th}$ segment with $N_i$ records

↳ gain of using an attribute to make a partition over D

BEST ATTRIBUTE : < GINIsplit INDEX

## OVERFITTING

- when a model begins to "memorize" training data rather than "learning" to generalize from a trend
- when the model try to model all the small nuances in the training data

The generated tree may overfit the training data

- Tree is too wide: too many branches, some may reflect anomalies due to noise or outliers
- Tree is too long: too many nodes along each branch

↓ SOLUTION

PRUNING

**PRE PRUNING**
↳ stop splitting when not statistically significant

**POST PRUNING**
↳ train a "fully grown" tree and remove branches afterwards

**INFORMATION GAIN**

$G(Gender) = E(T,F)$
$- \%M \times E(M+, M-)$
$- \%F \times E(F+, F-)$

F+ → nº de F que são T
F- → " " " " " F

**DECISION TREES DUVIDAS**

→ quando temos atributos que não são binários → calcular I.G.
↓
THRESHOLD

→ empates

→ ex: 8 (midpoints)

## Exercise 1

| A | B | C | Classe |
|---|---|---|---|
| 0 | 0 | 2 | P |
| 0 | 1 | 8 | P |
| 1 | 0 | 6 | N |
| 1 | 1 | 7 | N |
| 1 | 1 | 3 | P |

a) ENTROPY

$$E(P,N) = -\frac{P}{P+N}\log_2\frac{P}{P+N} - \frac{N}{P+N}\log_2\frac{N}{P+N}$$

$$= -\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5} \simeq$$

$$\simeq 0{,}442179 + 0{,}52877 \simeq 0{,}970949$$

b) ATTRIBUTE TESTED IN THE ROOT? (C4.5)
↳ use INFORMATION GAIN

$$G(A) = E(S) - \overset{\%0}{\frac{2}{5}} \times E(0+,0-) - \overset{\%1}{\frac{3}{5}} \times E(1+,1-)$$

$$= E(P,N) - \frac{2}{5} \times E(2,0) - \frac{3}{5} \times E(1,2) =$$

$$= 0{,}970949 - 0 - \frac{3}{5} \times 0{,}918295 = 0{,}419972$$

$$E(2,0) = -\frac{2}{2}\log_2\frac{2}{2} - 0 = -1 \times 0 = 0$$

$$E(1,2) = -\frac{1}{3}\log_2\left(\frac{1}{3}\right) - \frac{2}{3}\log_2\left(\frac{2}{3}\right) =$$

$$\simeq 0{,}52832 + 0{,}389975 = 0{,}918295$$

$$G(B) = E(P,N) - \frac{2}{5} \times E(0+,0-) - \frac{3}{5} \times E(1+,1-) =$$

$$= E(P,N) - \frac{2}{5} \times E(1,1) - \frac{3}{5} \times E(2,1)$$

$$= 0{,}970949 - \frac{2}{5} \times 1 - \frac{3}{5} \times 0{,}918295 = 0{,}019972$$

$E(1,1) = -\frac{1}{2}\log_2\left(\frac{1}{2}\right) - \frac{1}{2}\log_2\left(\frac{1}{2}\right) =$

$E(2,1) = -\frac{2}{3}\log_2\left(\frac{2}{3}\right) - \frac{1}{3}\log_2\left(\frac{1}{3}\right) \simeq$

$0{,}389975 + 0{,}52832 = 0{,}918295$

| A | B | C | Classe | C' |
|---|---|---|---|---|
| 0 | 0 | 2 | P | 1 |
| 1 | 1 | 3 | P | 1 |
| 1 | 0 | 6 | N | 0 |
| 1 | 1 | 7 | N | 0 |
| 0 | 1 | 8 | P | 0 |

midpoint : $\frac{2+8}{2} = \frac{10}{2} = 5$

$<= 5 \rightarrow$ (1) , $> 5 \rightarrow$ (0)

$G(C) = E(P,N) - \frac{3}{5} \times E(0+,0-) - \frac{2}{5} \times E(1+,1-)$

$= E(P,N) - \frac{3}{5} \times E(1,2) - \frac{2}{5} \times E(2,0)$

$= 0{,}970949 - \frac{3}{5} \times 0{,}918295 =$

$= 0{,}419972$

ATTRIBUTE TESTED IN THE ROOT $\rightarrow$ (C) ou (A) ?

$\text{GainRatio}(A) = \frac{0{,}419972}{2{,}05889} = 0{,}2039798$

$\text{splitInfo} = -\frac{2}{5}\log_2\left(\frac{2}{5}\right) - \frac{3}{5}\log_2\left(\frac{3}{5}\right) =$

$= 2{,}05889$

$\text{GainRatio}(C) = \frac{0{,}419972}{2{,}321928} = 0{,}18087$

$\text{splitInfo}(0,1) = -\frac{1}{5}\log_2\left(\frac{1}{5}\right) - \frac{1}{5}\log_2\left(\frac{1}{5}\right)$

$- \frac{1}{5}\log_2\left(\frac{1}{5}\right) - \frac{1}{5}\log_2\left(\frac{1}{5}\right) - \frac{1}{5}\log_2\left(\frac{1}{5}\right)$

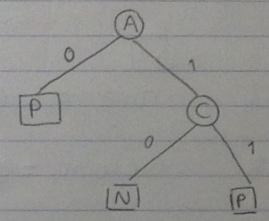
$= 2{,}321928$

ATTRIBUTE TESTED IN THE ROOT $\rightarrow$ $\boxed{A}$

c) $E(1,2) = -\frac{1}{3}\log_2\left(\frac{1}{3}\right) - \frac{2}{3}\log_2\left(\frac{2}{3}\right) = 0{,}918295$

quando A é (1), entropia da classe

G(B) e G(C) --- calcular e o > pica na folha




## Exercise 2

| A | B | C | Classe |
|---|---|---|---|
| 1 | 3 | 0 | P |
| 1 | 6 | 0 | N |
| 2 | 5 | 0 | N |
| 3 | 2 | 0 | N |
| 3 | 4 | 1 | P |
| 4 | 3 | 1 | N |
| 4 | 5 | 1 | N |
| 5 | 2 | 1 | N |
| 5 | 4 | 0 | N |
| 5 | 6 | 1 | N |

a) ENTROPY

$E(2,8) = -\frac{2}{10}\log_2\left(\frac{2}{10}\right) - \frac{8}{10}\log_2\left(\frac{8}{10}\right)$

$= 0{,}464385619 + 0{,}257542475 9 \simeq 0{,}7219$

b) $\text{split point}_A = 3$ $\quad \leq 3 \rightarrow$ (1) $\quad > 3 \rightarrow$ (0)

$\text{split point}_B = 4$ $\quad \leq 4 \rightarrow$ (1) $\quad > 4 \rightarrow$ (0)

## Exercise 8

| | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| X1 | 2 | 2.5 | 3 | 3.5 | 4 | 2 | 2.5 | 3 | 3.5 | 4 | 2 | 2.5 | 3 | 3.5 | 4 | 2 | 2.5 | 3 | 3.5 | 4 |
| X2 | 0,5 | 0,5 | 0,5 | 0,5 | 0,5 | 1 | 1 | 1 | 1 | 1 | 1,5 | 1,5 | 1,5 | 1,5 | 1,5 | 2 | 2 | 2 | 2 | 2 |
| C | N | N | N | N | N | N | N | N | N | N | N | N | P | P | P | N | N | P | P | N |

$\boxed{X1}$

$MIDPOINT_1 = \frac{2+2,5}{2} = 2,25$

$MIDPOINT_2 = \frac{2,5+3}{2} = 2,75$

$MIDPOINT_3 = \frac{3+3,5}{2} = 3,25$

$MIDPOINT_4 = \frac{3,5+4}{2} = 3,75$

$\boxed{X2}$

$MIDPOINT_1 = \frac{0,5+1}{2} = 0,75$

$MIDPOINT_2 = \frac{1+1,5}{2} = 1,25$

$MIDPOINT_3 = \frac{1,5+2}{2} = 1,75$

$Gini_{split}(X1 \leq 2,25) = \frac{4}{20} \times gini(X1 \leq 2,25 = T) + \frac{16}{20} \times gini(X_1 \leq 2,25 = F)$

$= \frac{16}{20} \times 0,4296875 \simeq 0,34375$

$gini(X1 \leq 2,25 = T) = 1 - \frac{4^2+0^2}{4^2} = 1 - 1 = 0$

$gini(X1 \leq 2,25 = F) = 1 - \frac{5^2+11^2}{16^2} = 0,4296875$

$Gini_{split}(X1 \leq 2,75) = \frac{8}{20} \times gini(X1 \leq 2,75 = T) + \frac{12}{20} \times gini(X1 \leq 2,75 = F)$

$= \frac{12}{20} \times 0,4861 = 0,29167$

$gini(X1 \leq 2,75 = T) = 1 - \frac{8^2}{8^2} = 0$

$gini(X1 \leq 2,75 = F) = 1 - \frac{5^2+7^2}{12^2} = 0,4861$

$Gini_{split}(X2 \leq 1,25) = \frac{10}{20} \times gini(X2 \leq 1,25 = T) + \frac{10}{20} \times gini(X2 \leq 1,25 = F)$

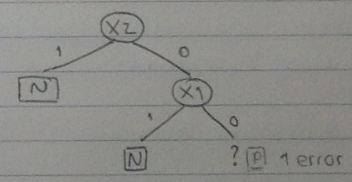
$= \frac{1}{2} \times 0,5 = 0,25$

$gini(X2 \leq 1,25 = T) = 1 - \frac{10^2}{10^2} = 0$

$gini(X2 \leq 1,25 = F) = 1 - \frac{5^2+5^2}{10^2} = 0,5$

ATTRIBUTE TESTED IN THE ROOT : $\boxed{X2}$

b)




c) $ACCURACY = \frac{TP+TN}{All} = \frac{19}{20}$

$SENSIBILITY = \frac{TP}{TP+FN} = \frac{5}{5} = 1$

$SPECIFICITY = \frac{TN}{TN+FP} = \frac{14}{15}$

## Exercise 13

a) D → for numeric. C4.5 and CART

b) B

c) 27 instances values (0,25 ; 0,5 ; 0,75)

$9N + 7N + 2N = 18N$

$2P + 6P + 1P = 9P$

$E(9,18) = -\frac{9}{27} \log_2\left(\frac{9}{27}\right) - \frac{18}{27} \log_2\left(\frac{18}{27}\right)$ C

$= 0,52832 + 0,389975 = 0,918$

d)

$$\text{Accuracy} = \frac{TP+TN}{All} = \frac{24}{27} = 0{,}89$$ (B)

e)

$$\text{sensibility} = \frac{TP}{TP+FN} = \frac{6}{9} = 0{,}67$$ → (C)

$$\text{specificity} = \frac{TN}{TN+FP} = \frac{15}{15} = 1$$

## Ensembles

**Bias**: systematic deviation of the predicted decision boundary from the true one
(depends on the adequacy of the classifier to the nature of data)

**Variance**: deviation among the learned decision boundaries over different training sets

Ensembles
- Bagging
- random forest

**Unstable classifiers**: small perturbations in the training set result in large changes in the prediction or decision boundary

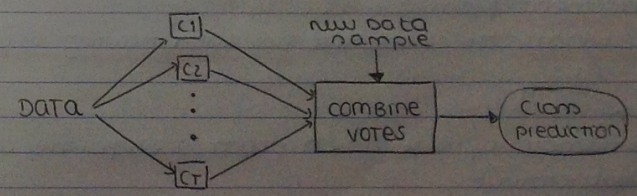
**High variance classifiers**: unstable (they tend to overfit the data)

**High bias classifiers**: underfit the data, usually have low variance

↓ Goal

Reduce classification error : reducing bias + variance

Ensemble method:
1) Learn several simple classifiers : Reduces variance
2) Combine the output of the simple classifiers to produce the classification decision : Reduces Bias




↳ divide-and-conquer approach
↳ the model improves accuracy and robustness

**DIVERSITY**: difference between classifier outputs
↳ if all classifiers are the same, no ensemble is needed
↳ when a classifier makes an error, other classifiers will complement it

**SAMPLING TRAINING EXAMPLES**: train K classifiers on K subsets drawn from the training set
↳ BOOTSTRAP SAMPLE + RE-WEIGHT DATA

**SAMPLING FEATURES**: train K classifiers on K subsets of features drawn from the feature space

**DIFFERENT LEARNING MODELS**: use all the training examples, but apply different learning algorithms

**LEARNING "RANDOMLY"**: introduce randomness into learning procedures

VOTING SCHEMES: (COMBINE CLASSIFIERS)
→ unanimous vote
→ majority vote
→ weighted majority vote

## BAGGING

- Given a set D of n training examples, at each iteration I, a training set Di of n examples is sampled with replacement from D - **BOOTSTRAP**
- A classifier model Mi is learned for each training set Di

- FOR i = 1..K
  - draw a sample of size $n^* < n$ from D uniformly and with replacement
  - learn classifier Mi

- Final classifier is a vote of M1...MK

**CLASSIFICATION**: classify an unknown sample X
↳ each classifier Mi returns its class prediction
↳ the bagged classifier M* counts the votes and assigns the class with the most votes to X

**PREDICTION**: can be applied to the prediction of continuous values by taking the average value of each prediction for a given test instance.

n completely independent binary classifiers
↓
**BINOMIAL DISTRIBUTION**
(the majority vote will be the optimal combination rule)
→ discrete probability distribution of the number of successes in a sequence of n independent yes/no experiments
→ it is used to model the number of successes in a sample of size n drawn with replacement from a population of size N

probability of getting exactly x successes in n trials

$X \sim B(n,p)$ $\quad P(X=x) = C(n,x)\, p^x (1-p)^{n-x}$

$$C(n,x) = \frac{n!}{x!(n-x)!}$$

- BAGGING makes the correct classification if at least **n/2 + 1** classifiers choose the correct label
(accuracy follows the binomial distribution)

$$\sum_{k=\frac{n}{2}+1}^{n} C_k^n p^k (1-p)^{n-k} \sim 1 \text{ as } n \to \infty \text{ and } p > 0,5$$

**ACCURACY**: decreases error by decreasing the variance in the results (due to unstable learners)
↓
algorithms whose output can change dramatically when the training data is slightly changed

↳ uses RESAMPLING rather than reweighting
↳ does not modify the distribution over examples or mislabels
↳ uses uniform distribution
↳ it gives equal weight

## RANDOM FORESTS

→ each classifier in the ensemble is a decision tree classifier
  ↳ is generated using a random selection of attributes at each node to determine the split
→ each tree votes
→ the most popular class is returned

CONSTRUCTION

1) FOREST-RI (random input selection)
   ↳ randomly select, at each node, F attributes as candidates for the split at the node
   (CART used to grow the trees to maximum size)

2) FOREST-RC (random linear combinations)
   ↳ creates new attributes (or features) that are a linear combination of the existing attributes
   (reduces the correlation between individual classifiers)

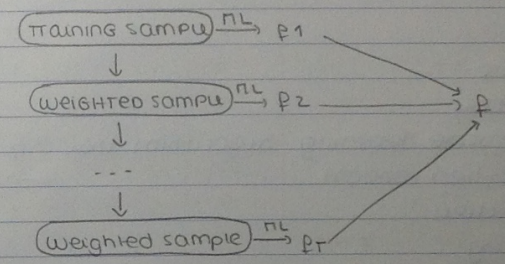
ALGORITHM (repeat K times)

1) draw with replacement N examples, put in train set
2) build d-tree, but in each recursive call
   - choose (w/o replacement) i features
   - " best of these i as the root of this (sub)tree
3) do NOT prune

- improve accuracy
- " efficiency
- robust to errors and outliers
- deal with large # of features
- reduce overfitting substantially

## BOOSTING

- perform iterative search to locate the regions/examples that are more difficult to predict
- through each iteration reward accurate predictions on those regions
- combines the rules from each iteration




→ weights are assigned to each training instance
→ a series of K classifiers is iteratively learned
→ after a classifier $M_i$ is learned, the weights are updated to allow the subsequent classifier, $M_{i+1}$, to pay more attention to the training instances that where misclassified by $M_i$
→ $M^*$ combines the votes of each individual classifier, where the weight of each classifier's vote is a function of its accuracy

output of the Adaboost algorithm

$$M^*(x) = \text{sgn}\left[\sum_{i=1}^{K} \alpha_i M_i(x)\right] \qquad \alpha_i = \frac{1}{2}\log\left(\frac{1-err_i}{err_i}\right)$$

$$err_i = \sum_{j=1}^{n} w_j^{(i)}\, h(-y_j M_i(x_j)) \leq 1/2 - \varepsilon \qquad (\varepsilon \text{ small})$$

- $w_j = \frac{1}{N}$ → err = classical error rate
  ↓ 1 − accuracy
- this is true for $i=0$

Training Error

$$TE = \frac{\sum_{j=1}^{N} (H^*(x_j) \neq y_j)}{N}$$

$$TE \leq \prod_{i=1}^{K} \left( \sum_{j=1}^{N} w_j^{(i)} e^{\alpha_i y_j h_i(x_j)} \right)$$

$$TE \leq e^{-2K\gamma^2} \quad , \gamma_i = 1/2 - err \qquad \gamma_i > \gamma > 0$$

TE decreases exponentially with K

→ no parameters to tune
→ fast, simple and easy
→ focus on finding base learning algorithms that only need to be better than random
→ it can identify outliers
→ it resists overfitting

→ susceptible to noise → causes overfitting

→ can fail to perform when
- insufficient data
- weak learners are too weak

Exercise 1

Treestumps → decision tree with only one binary node

| A | B | C | Classe | |
|---|---|---|---|---|
| 0 | 0 | 0.25 | N | |
| 0 | 1 | 0.25 | P | |
| 1 | 0 | 0.25 | P | |
| 1 | 1 | 0.25 | N | |
| 0 | 0 | 0.5 | N | |
| 0 | 1 | 0.5 | P | |
| 1 | 0 | 0.5 | P | |
| 1 | 1 | 0.5 | N | |
| 0 | 0 | 0.75 | N | ← x |
| 0 | 1 | 0.75 | P | |
| 1 | 0 | 0.75 | P | |
| 1 | 1 | 0.75 | N | ← x |

1)

A
- 0: 4P+2N
- 1: 4P+2N

B
- 0: 4P+2N
- 1: 4P+2N

C < 0,75
- 1: 4N+4P
- 0: ?

não há maioria

2) (1,1,0.4)

Isn't able to classify → ?

Ties

Exercise 2

Regular trees with just two levels of nodes

?

## Exercise 3

Two weak models:

1) accuracy = 75% → positive
2) accuracy = 66% → negative

$$M^*(x) = \text{sgn}\left[\sum_{i=1}^{2} \alpha_i M_i(x)\right]$$

$$\alpha_1 = \frac{1}{2}\log\left(\frac{1-0{,}25}{0{,}75}\right) = \cancel{0{,}2385}$$

$$\alpha_2 = \frac{1}{2}\log\left(\frac{1-0{,}34}{0{,}66}\right) = \cancel{0{,}144}$$

$M_1 \rightarrow \text{sgn}(+0{,}2385 - 0{,}144) \rightarrow$ POSITIVO

$$\text{sgn} \begin{cases} >0 & \rightarrow 1 \\ <0 & \rightarrow -1 \end{cases}$$

A

## Association Rules (pattern mining)

→ find frequent patterns, associations, correlations or causal structures among sets of items or objects

→ Given:

1) database of transactions (each transaction is a list of items purchased by a customer in a visit)
2) a minimum support threshold (min_sup)
3) " " confidence " (min_conf)

ASSOC RULES:
- APRIORI
- ECLAT
- FP-GROWTH

→ Find ALL rules:

- $A \rightarrow B$ (A and B sets of items)
- $sup \geq min_sup$
- $conf \geq min_conf$

**SUPPORT (sup)** → probability that a transaction contains the elements in A and the elements in B

$$sup(A \Rightarrow B) = P(A \cap B)$$ utility!

**CONFIDENCE (conf)** → conditional probability that a transaction that contains the items in A also contains the items in B

$$conf(A \Rightarrow B) = P(B|A) = \frac{P(A \cap B)}{P(A)}$$ precision!

**ITEMSET SEARCH SPACE**: lattice where any two itemsets A and B are connected by a link iff $A \subset B$ and $|A| = |B| - 1$
(Breath-first or Depth-first)

**MONOTONICITY**: if for any pattern A satisfying C, every super-pattern of A also satisfies it

**ANTI-MONOTONICITY**: if a pattern A does not satisfy C, none of the super-patterns of A can satisfy C
support ↗

- Find FREQUENT ITEMSETS (itemsets with enough support).
  - → Apriori
  - → FP-growth
  - → ECLAT

↳ used to generate ASSOCIATION RULES with enough confidence
  → genrules

## APRIORI

L1 Finding
$\boxed{L1}$ → frequent itemsets

CANDIDATE GENERATION
$C_k \leftarrow L_{k-1} \times L_{k-1}$
$L_k$
Prune with Anti-monotonicity
$\boxed{C_k}$
SUPPORT PRUNING

### CANDIDATES GENERATION

1) SELF-JOINING $L_{k-1}$

$A = (a_1, a_2, \dots, a_{k-1})$ $B = (b_1, b_2, \dots, b_{k-1})$

if $a_1 = b_1$ and $a_2 = b_2$ ... and $a_{k-1} < b_{k-1}$

Then $AB = (a_1, a_2, \dots, a_{k-1}, b_{k-1}) \in C_k$

ex $(abc), (abd), (acd) \in L3 \rightarrow (abc) \times (abd) = (abcd)$

2) PRUNING

```
forall itemsets c ∈ Ck do
    forall (k-1)-subsets s of c do
        if s ∉ Lk-1 then
            delete from Ck
```

ex: $(abc), (abd), (acd) \in L3$
$(abc) \times (abd) = (abcd)$
but $(bcd) \notin L3 \rightarrow (abcd) \notin L4$

### SUPPORT COUNTING

- candidate itemsets are stored in a hash-tree
- leaf node of hash-tree contains a list of itemsets and counts
- Interior node contains a hash table
- subset function : finds all the candidates contained in a transaction

- use frequent (k-1)-itemsets to generate candidate frequent k-itemsets
- use database scan and pattern matching to collect counts for the candidate itemsets

## ECLAT (Equivalence Class Transformation)

→ combines the lists of transactions (tidsets) that support each pair of k-patterns that share the same prefix, in a depth-first manner

$t(AB) = t(A) \cap t(B)$ (A: tidset, B: tidset) A and B frequent itemsets

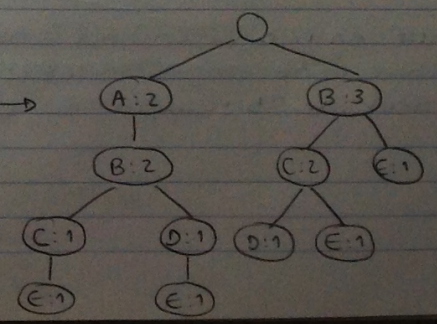
support $= |t(XY)|$

## FP-GROWTH

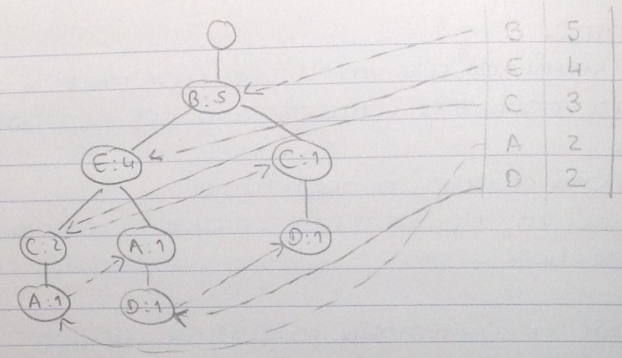
→ compress a large database into a compact, FP-tree structure

- divide-and-conquer methodology : decompose mining tasks into smaller ones
- avoid candidate generation : sub-database test only!

ABCE
BCD
ABDE
BCE
BE




| | |
|---|---|
| B | 5 |
| E | 4 |
| C | 3 |
| A | 2 |
| D | 2 |




- for each item, construct its conditional pattern-base, and then its conditional FP-tree
- repeat the process on each newly created conditional FP-tree
- until the resulting FP-tree is empty, or it contains only one path

① scan DB once, find frequent 1-itemset (single item pattern)
② order frequent items in frequency descending order
③ scan DB again, construct FP-tree

**SEQUENTIAL PATTERN MINING** (SPaM)
↳ address the problem of discovering the existing maximal frequent sequences in a given database

- A sequence $a_1a_2 \dots a_n$ is a **SUBSEQUENCE** of another sequence $b_1b_2 \dots b_m$ if there exist integers $i \leq i_1 \leq \dots \leq i_n \leq m$ such that $a_1 \subseteq b_{i_1}$, $a_2 \subseteq b_{i_2}$, ..., $a_n \subseteq b_{i_n}$

- A sequence $a_1a_2 \dots a_n$ is a **δ-distance subsequence** of $b_1b_2 \dots b_m$, if there exist integers $1 \leq i_1 \leq i_2 < \dots < i_n \leq m$ such that $a_1 \subseteq b_{i_1}$, $a_2 \subseteq b_{i_2}$, ..., $a_n \subseteq b_{i_n}$ and $i_k - i_{k-1} \leq \delta$

**MAXIMAL SEQUENTIAL PATTERN**: not a subsequence of a seq. pattern.

**CLOSED SEQUENTIAL PATTERN**: not a subsequence of a seq. pattern with same support.

1) **GSP** (Apriori-based)
→ follow the candidate gene
→ Breath-first traversal algor

2) **PREFIXSPAN** (Pattern-growth)
→ avoid candidate generation
→ depth-first traversal algorithms
→ restrict the search space at each step

X is a **MAX-PATTERN** if it is frequent and there exists no frequent supper-pattern $Y \supset X$

X is **CLOSED** if it is frequent and there exists no supper-pattern $Y \supset X$, with the same support as X
↳ reduces the # of patterns and rules

**CONSTRAINED PATTERN MINING**: the discovery of the complete set of patterns satisfying C.
(constraint: predicate over sequences)

Measures:
1) **LIFT** (correlation, interest)
↳ ratio of the observed joint probability of X and Y to the expected joint probability if they were statistically independent. Measures the surprise of the rule

$$\text{lift}(A \Rightarrow B) = \frac{P(B|A)}{P(B)} = \frac{P(A \cap B)}{P(A) \times P(B)}$$

- $P(A \cap B) = P(A) \times P(B) \Leftrightarrow$ A and B are independent → LIFT = 1
- A and B negatively correlated → LIFT < 1
- " " " positively " → LIFT > 1

→

2) Leverage

↳ measures the ≠ between the observed and expected joint probability of XY assuming that X and Y are independent

$$\text{leverage}(X \to Y) = P(XY) - P(X) \cdot P(Y)$$

- $P(A \cap B) = P(B) \times P(A) \to$ Leverage = 0
- A and B negatively correlated $\to$ Leverage < 0
- " " " positively " $\to$ Leverage > 0

3) JACCARD

↳ measures the similarity between two sets

↳ it computes the similarity between the transactions supporting X and Y

$P(A \cap B) = P(A \cup B) \to$ JACCARD = 1

$P(A \cap B) < P(A \cup B) \to$ JACCARD < 1

Association rules

Quizzes

→ How many time does Apriori scan the dataset?

patterns → maximal and frequent

Exercise 1

min_support → 40 $= \frac{16}{10} = 1,6$

Items: ACD, BCE, A

a) APRIORI: determin ... ets

C1:

{A} 2/4
{B} 3/4
{C} 3/4
~~{D} 1/4~~
{E} 3/4

{A} 2/4
{B} 3/4
{C} 3/4
{E} 3/4

C2: pruning

~~{A,B} 1/4~~
{A,C} 2/4
~~{A,E} 1/4~~
{B,C} 2/4
{B,E} 3/4
{C,E} 2/4

L2:

{A,C} 2/4
{B,C} 2/4
{B,E} 3/4
{C,E} 2/4

C3:

{B,C,E} 2/4

L3

{B,C,E} 2/4

C4: $\emptyset$

b) ASSOCIATION RULES WITH CONFIDENCE > 50%.

{B,C,E}

$CONF(X \to Y) = P(Y|X) = \frac{P(Y \cap X)}{P(X)}$

| | X | Y | |
|---|---|---|---|
| L3 | {B,C} | → E | 2/2 = 1 |
| | {B,E} | → C | 2/3 = 0,6 |
| | {C,E} | → B | 2/2 = 1 |
| | B | → {C,E} | 2/3 |
| | C | → {B,E} | 2/3 |
| | E | → {B,C} | 2/3 |

$conf(X \to Y) = P(Y|X) = \frac{P(Y \cap X)}{P(X)}$

C → candidates

L → frequent item sets

L2

| X | | Y | | X | | Y | |
|---|---|---|---|---|---|---|---|
| {A} | → | C | 2/2 = 1 | A | → | {C} | 2/2 = 1 |
| {B} | → | C | 2/3 | B | → | {C} | 2/3 |
| {B} | → | E | 3/3 | B | → | {E} | 3/3 |
| {C} | → | E | 2/3 | C | → | {E} | 2/3 |

## Exercise 2

$\frac{40}{100} \times 5 = \frac{20}{10} = 2$ Σ

a)

C1:

{A} 3/5
{B} 3/5
{C} 3/5
{D} 4/5
{E} 2/5
~~{F} 1/5~~
~~{G} 1/5~~

L1:

{A} 3/5
{B} 3/5
{C} 3/5
{D} 4/5
{E} 2/5

C2:

{A,B} 2/5
~~{A,C} 1/5~~
{A,D} 2/5
~~{A,E} 1/5~~
{B,C} 2/5
{B,D} 2/5
{B,E} 2/5
{C,D} 3/5
~~{C,E} 1/5~~
~~{D,E} 1/5~~

L2:

{A,B} 2/5
{A,D} 2/5
{B,C} 2/5
{B,D} 2/5
{B,E} 2/5
{C,D} 3/5

C3:

{A,B,D}
{B,C,D}
{B,D,E}
{B,C,E}

A

b)

$conf(A \to D) = P(D|A) = \frac{P(D \cap A)}{P(A)} = \frac{2}{3}$

$conf(D \to A) = P(A|D) = \frac{P(A \cap D)}{P(D)} = \frac{2}{4} = \frac{1}{2}$

$LIFT(A \to D) = \frac{P(D \cap A)}{P(D) \times P(A)} = \frac{\frac{2}{5}}{\frac{3}{5} \times \frac{4}{5}} = \frac{\frac{2}{5}}{\frac{12}{25}} = \frac{2 \times 25}{5 \times 12} = \frac{50}{60} = \frac{5}{6}$

$LIFT(D \to A) = \frac{P(A \cap D)}{P(A) \times P(D)} = \frac{5}{6}$

A

## Exercise 3

ANTI-MONOTONY

SUP(3) = 2/5
SUP(1) = 3/5
SUP(2) = 3/5
SUP(4) = 4/5
SUP(5) = 2/5
~~SUP(6) = 1/5~~
~~SUP(7) = 1/5~~

L1:

{1}
{2}
{3}
{4}
{5}

C2:

{1 2} 2/5
~~{1 3}~~ 1/5
{1 4} 2/5
~~{1 5}~~ 1/5
{2 3} 2/5
{2 4} 2/5
{2 5} 1/5
{3 4} 3/5
~~{3 5}~~ 1/5
~~{4 5}~~ 1/5

C 235, 245
23 ✓
35 not appears 1 (in item superset minimal)

## Exercise 7

{A, B, C, D, E}  2 patterns: ABCD, ACDE

a) B

b) A

L3 = {ABD, ABE, ACD, ADE, BCE, CDE, BDE}
L2 = {AB, AD, AE, AC, BC, BE, CD, CE, BD}

Exercise 8 ABDE, ACDE and BCE

C3 = {ABD, ABE, ABC, ADE, ADC, AEC, BCE, ...}

{ADEBC}

a) B
b) A

ADEB
ADEC
DEBC

Exercise 10 ABCD, ABDE, DF

a)// FREQUENT ITEMSETS
b)

PATTERN → frequent and maximal

① {A}, {B}, {C}, {D}, {E}, {F}
② {AB} {AC} {AD} {AE} {AF} {BC} {BD} {BE} {BF} {CD} {CE} {CF} {DE} {DF} {EF}
③ {ABC} {ABD} {ABE} {ABF}
...

c) ABCD ABDE DF

④

Exercise 11
ABCD ABCE ABDE ACDE

④

ABCDE

ABCD ABCE BCDE ACDE ABDE

pelo anti-monotonicity

Exercise 12
ABCD and ABCE

a) Apriori algorithm scans the dataset 4 times (because there is a candidate (ABCDE) with 5 items, it has a subpattern (BCDE) that is non-frequent and so the candidate is eliminated by the anti-monotonic property.

b) calcular L3 → L2 → C3, ver se C3 = L3

## SOCIAL NETWORK ANALYSIS

**WEB MINING**: use of data mining techniques to automatically discover and extract information from web documents and services

WEB USAGE MINING — to find usage patterns
↳ to identify user profiles

WEB CONTENT MINING ↔ TEXT and multimedia DM
↳ to find info in the existing data in web documents

SNA
- PAGERANK
- HITS

WEB STRUCTURE MINING ↔ SOCIAL NETWORK ANALYSIS
↳ to find the model underlying the link structures
↳ to compare web sites

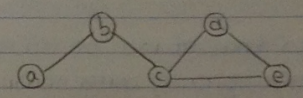
**NETWORK**: collection of entities that are interconnected with links.

**SOCIAL NETWORK**: social structure of people, related to each other through a common relation

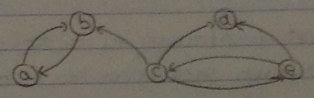
How the patterns of relations among actors affects individual behavior or system properties?

---

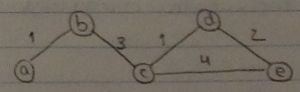
**SOCIAL NETWORK**
- represented as a **graph** G = (V, E)
  - **NODES**: individuals (defined by its label)
  - **EDGES**: relationships (defined by 2 nodes which it connects)
    ↳ directed and/or weighted



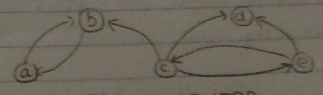
UNDIRECTED, UNWEIGHTED



DIRECTED, UNWEIGHTED



UNDIRECTED, WEIGHTED



DIRECTED, WEIGHTED

- 2 nodes are ADJACENT if they are connected by an edge

DEGREE: number of its edges
- IN-DEGREE: number of incoming edges
- OUT-DEGREE: number of outgoing edges

- Graphs may be:
  * UNIPARTITE: all nodes belong to a unique category
  * MULTIPARTITE: there are multiple categories of nodes, edges are only defined among nodes of ≠ categories

ADJACENCY MATRIX
- n nodes need a $n \times n$ matrix
- unweighted graphs: each cell in the matrix is 1 if the two nodes are connected; or 0 if they are not
- weighted graphs: each cell in the matrix corresponds to the weight for each weight

PATH / WALK: exists if you can traverse along edges from one node to another. (ordered list of nodes)
  LENGTH: number of edges in the path
  CYCLE: a path that starts and finishes at the same node

---

CONNECTED COMPONENT: set of nodes and edges where there exists a path between any 2 nodes in the set
  WEAKLY CONNECTED COMPONENT: replacing directed by undirected edges, there is a path between any 2 nodes
  STRONGLY CONNECTED COMPONENT: there is a directed path between any 2 nodes

DISTANCE: minimum number of edges that must be traversed on the path from one node to another, ignoring directionality

DIAMETER (LONGEST SHORTEST PATH): maximum distance between any two nodes

EFFECTIVE DIAMETER: distance at which 90% of all connected pairs of nodes can be reached

GEODESIC DISTANCE (SHORTEST PATH): number of nodes in the shortest possible walk from one node to another.

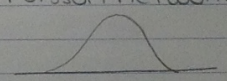
"small-world phenomenon" — web >> 19
"six degrees of separation" — social networks >> 6

DEGREE DISTRIBUTIONS
→ degrees in real networks are heavily skewed to the right
→ distribution has a long tail of values that are far above the mean

fraction of nodes with degree k

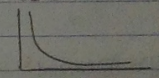
POWER-LAW: $p_k \sim k^{-\alpha}$
↳ line on log-log plot

Poisson network



degree distribution: Poisson

scale-free network



degree distribution: Power-law

NETWORK RESILIENCE: observe how the connectivity (length of the paths) of the network changes as the nodes get removed

EVOLVING GRAPHS:
- constant average degree: the number of edges grows linearly with the number of nodes
- slowly growing diameter: as the network grows the distances between nodes grow

DENSIFICATION
↳ networks are denser over time

# edges grows faster than # nodes

$E(t) \propto N(t)^{\alpha}$

average degree is increasing

densification exponent
= 1: linear growth

## SHRINKING DIAMETERS

- diameter stabilizes over time
- as the network grows the distances between nodes slowly decrease

---

## CENTRALITY

- important or prominent nodes are those that are linked or involved with other nodes extensively
  central node: involved in many ties

## DEGREE CENTRALITY

n: number of nodes

*UNDIRECTED GRAPH

$$C_D(i) = \frac{d(i)}{n-1}$$

degree of the node

*DIRECTED GRAPHS

$$C_O(i) = \frac{d_O(i)}{n-1}$$

out-degree

## CLOSENESS CENTRALITY

↳ a node is central if it can easily interact with all other nodes. its distance to all other nodes is short.

$$C_C(i) = \frac{n-1}{\sum_{j=1}^{n} d(i,j)}$$

↳ distance from node i to node j

⚠ For connected graphs

- For Directed Graph distance needs to consider directions of edges

**PRESTIGE:** a prestigious node is one who is object of extensive ties as a recipient

CENTRALITY → focus on OUT-LINKS
PRESTIGE → " " IN-LINKS

Rank prestige
↓
basis of HITS and PageRank algorithms

## DEGREE PRESTIGE

↳ degree prestige of a node i only considers the nodes that are adjacent to i

$$P_D(i) = \frac{d_I(i)}{n-1}$$

→ in-degree of i

## PROXIMITY PRESTIGE

closeness / distance of other nodes to i

$$P_P(i) = \frac{|I_i| / (n-1)}{\left(\sum_{j \in I_i} d(i,j) / |I_i|\right)}$$

→ average distance

↳ it is the ratio of nodes who can reach i to the average distance that these nodes are from i

$I_i$ (influence domain of i): the set of nodes that can reach node i.

Every node can reach node i → $P(i) = 1$

## RANK PRESTIGE

$$P_R(i) = A_{1i} P_R(1) + A_{2i} P_R(2) + \cdots + A_{ni} P_R(n)$$

↳ linear combination of links that point to i

$$P = A^T P$$

↳ vector that contains the rank prestige values for each node

↳ adjacency matrix of the network or graph

P is an EIGENVECTOR of $A^T$

→

## PAGERANK ALGORITHM (STATIC RANKING)

→ interprets a hyperlink from page x to page y as a vote, by page x, for page y

→ votes casted by "important" pages weight more heavily and help to make other pages more "important"

more in-links page i receives
↓
more prestige page i has

→ a page is important if it is pointed to by other important pages → FIGHTING SPAM

## PAGERANK SCORE (importance of page i)

$$P(i) = \sum_{(i,j)\in E} \frac{P(j)}{O_j}$$

$O_j$ → number of out-link of j

↓

sum of the PageRank scores of all pages that point to i

## ADJACENCY MATRIX OF THE GRAPH

TRANSITION PROBABILITY

$$A_{ij} = \begin{cases} \frac{1}{O_i} & \text{if } (i,j) \in E \\ 0 & \text{otherwise} \end{cases}$$

A is a stochastic matrix
A is irreducible
A is aperiodic
} we can use power iteration to find P

## TRANSITION PROBABILITY MATRIX

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ \vdots & \vdots & & \vdots \\ A_{n1} & A_{n2} & \cdots & A_{nn} \end{bmatrix} = \begin{bmatrix} 1/O_1 & 1/O_1 & \cdots & 1/O_1 \\ \vdots & \vdots & & \vdots \\ 1/O_n & 1/O_n & \cdots & 1/O_n \end{bmatrix}$$

$A_{ij}$: represents the transition probability that the surfer in page i will move to page j.

no out-links : DANGLING PAGES → A is not stochastic

IF $\sum_{j=1}^{n} A_{ij} = \sum_{j=1}^{n} 1/O_i = 1$ → A would be a stochastic matrix

Fix the problem → add a complete set of outgoing links from each such page i to all the pages on the web

A is not IRREDUCIBLE
↓
web graph G is STRONGLY CONNECTED
↓
for each pair of nodes $u, v \in$ vertices, there is a path from u to v

A is not APERIODIC (k = 1)
↓
PERIODIC: (period k > 1) if k is the smallest number such that all paths leading from state i back to state i have a length that is a multiple of k

↓ turn IRREDUCIBLE + APERIODIC

## PAGERANK SCORE

$$P(i) = (1-d) + d \underbrace{\sum_{(j,i)\in E} \left(\frac{P_j}{O_j}\right)}_{= \; d \sum_{j=1}^{n} A_{ji} P(j)}$$

$(1-d)$ → damping factor

Probability d: randomly chooses an out-link to follow
" d-1: jumps to a random page

HYPERTEXT INDUCED TOPIC SEARCH

## HITS Algorithm (search query dependent)

AUTHORITY: page with many in-links.
HUB: page with many out-links

authorities and hubs have a mutual reinforcement relationship
↓
create a BIPARTITE SUB-GRAPH

**COLLECTION OF PAGES:**

→ sends the query q to a search engine
→ collects t highest ranked pages → this set is called the ROOT set W
→ grows W by including any page pointed to by a page in W and any page that points to a page in W. This gives a larger set S: Base set

- HITS works on the pages in S and assigns every page in S an authority score and a hub score

adjacency matrix

$$L_{ij} = \begin{cases} 1 & \text{if } (i,j) \in E \\ 0 & \text{otherwise} \end{cases}$$

⊕ provides more relevant authority and hub pages

⊖ - easily spammed
- topic drift
- inefficiency at query time

**HITS scores**

$$a(i) = \sum_{(j,i) \in E} h(j) \qquad h(i) = \sum_{(i,j) \in E} a(j)$$

authority score of the page i — hub score of the page i

matrix form

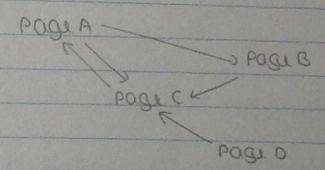
a → column vector with all authority scores
h → " " " " " hub "

$$\vec{h} \leftarrow A\vec{a} \qquad \vec{h} \leftarrow AA^T\vec{h}$$
$$\vec{a} \leftarrow A^T\vec{h} \quad (\Longleftrightarrow) \quad \vec{a} \leftarrow A^TA\vec{a}$$

→ uses POWER ITERATION

## Social Network analysis

**Exercise 1**




a) Graph diameter
↳ > distância entre nós

③ (DAB)

b) Degree, In-Degree, Out-Degree
↳ quantos nós vão p ele — ↳ quantos nós saem ele

| | In-Degree | Out-Degree | Degree? |
|---|---|---|---|
| A | 1 | 2 | 3 |
| B | 1 | 1 | 2 |
| C | 3 | 1 | 4 |
| D | 0 | 1 | 1 |

c) Degree centrality

$C_D(A) = \frac{2}{3}$

$C_D(B) = \frac{1}{3}$

$C_D(C) = \frac{1}{3}$

$C_D(D) = \frac{1}{3}$

Closeness centrality

$C_C(A) = \frac{3}{1+1+1} = \frac{3}{2}$

$C_C(B) = \frac{3}{2+1} = 1$

$C_C(C) = \frac{3}{\ }$

$C_C(D) = \frac{3}{\ }$

**Exercise 2** ← ver teste 4 solução

PageRank algorithm

a) Initialization for the PageRank scores
↳ prob. uniforme entre os nós
$\frac{1}{4}$

b) Transition probability matrix

$$A = \begin{array}{c} \begin{array}{cccc} A & B & C & D \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 \\ 1/3 & 1/3 & 1/3 & 0 \\ 0 & 0 & 1/2 & 1/2 \end{bmatrix} \end{array}$$

+ links - importante

c)

$$PR = \underbrace{1-d}_{\text{damping factor}} + d \times \sum \left( \frac{PR \text{ pag. ligadas ao nó (inlinks)}}{\#\text{arcos que saem do nó}} \right)$$

PageRank (A) $= 1 - 0{,}8 + 0{,}8 \times \left( \frac{1/4}{3} \right) = 0{,}267$

PageRank (B) $= 1 - 0{,}8 + 0{,}8 \times \left( \frac{1/4}{1} + \frac{1/4}{3} \right) = 0{,}467$

PageRank (C) $= 1 - 0{,}8 + 0{,}8 \times \left( \frac{1/4}{2} + \frac{1/4}{2} + \frac{1/4}{3} \right) = 0{,}467$

PageRank (D) $= 1 - 0{,}8 + 0{,}8 \times \left( \frac{1/4}{2} + \frac{1/4}{2} \right) = 0{,}4$

## Exercise 3

a) $\frac{1}{5}$

b)

$$P\Pi = \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \end{array} \begin{array}{c} \begin{array}{ccccc} A & B & C & D & E \end{array} \\ \begin{bmatrix} 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1/2 & 0 & 0 & 1/2 \end{bmatrix} \end{array}$$

c)

PageRank (A) $= 1 - 0{,}7 + 0{,}7 \times \left( \frac{1/5}{5} \right)$

PageRank (B) $= 1 - 0{,}7 + 0{,}7 \times \left( \frac{1/5}{5} + \frac{1/5}{2} \right)$

PageRank (C) $= 1 - 0{,}7 + 0{,}7 \times \left( \frac{1/5}{1} + \frac{1/5}{5} \right)$

PageRank (D) $= 1 - 0{,}7 + 0{,}7 \times \left( \frac{1/5}{5} + \frac{1/5}{1} \right)$

PageRank (E) $= 1 - 0{,}7 + 0{,}7 \times \left( \frac{1/5}{5} + \frac{1/5}{1} + \frac{1/5}{2} \right)$

Exercise 1

## FEATURE SELECTION (DATA TRANSFORMATION)

+ features → - performance
+ dimensionality → training observations required increases exponentially

↓

REDUCE DIMENSIONALITY

### CHALLENGES

- learning complexity
- generalization difficulty
  1) OVERFITTING : unability to discard non-informative or non-discriminative regions (global learning : sup + unsup)
  2) UNDERFITTING : exclusion of informative or discriminative regions from the learning (pattern-based learning : division trees, associative classifiers)

     feature or selection

     ↓ evaluate the risks

     → compare errors on training data vs testing data
     → analyse variability of errors accross testing folds

GLOBAL LEARNING : fit class-conditional multivariate distribution to describe / classify data

LOCAL LEARNING : choose few attributes as long as their combination of values discriminate a class

### DIMENSIONALITY REDUCTION

↳ project the m-dimensional points into a k-dimensional space ($k \ll m$), preserving most of relevant information or structure from data

1) FEATURE SELECTION : choose a subset of all features

$$[y_1, y_2, \dots, y_m] \rightarrow [y_{i1}, y_{i2}, \dots, y_{ik}]$$

2) FEATURE EXTRACTION : creating new features by combining existing ones

$$[y_1, y_2, \dots y_m] \rightarrow [a_1, a_2, \dots a_k] = f([y_{i1}, y_{i2}, \dots, y_{im}])$$

3) SPACE KERNELS : exclude non-relevant features
   - SVM, discriminant analysis, parametric mixtures

4) SUBSPACE SELECTION

- pattern mining and biclustering
- decision trees and random forests
- associative classifiers and decision tables

↓ GOAL

→ guide unsupervised learning (informative regions)
→ " supervised " (discriminative " )
→ visualization
→ data compression
→ noise removal
→ speed-up learning
→ simplicity and comprehensibility of mined results
→ map multimedia data

## FEATURE SELECTION

↳ choose an optimal subset of features according to an objective function

- minimize information loss
- maximize class discrimination

* FEATURE RANKING

↳ weighting each individual feature according to objective criteria (info gain)

↳ selecting top-ranked features
- threshold
- k-top
- percentile

⊕
→ efficient $O(m)$
→ no need to test combination of features

⊖
- hard to determine threshold or k
- unable to consider correlation between features

* OPTIMAL SUBSET OF FEATURES

- STARTING POINTS: empty set, full set, random set of features
- SEARCH STRATEGIES: heuristic, exhaustive (complete), non-deterministic

- SEQUENTIAL FORWARD SELECTION (SFS): start with $F = \emptyset$, and iteratively add the most relevant feature regarding F
  Generalized SFS: add p optimal features per iteration
  ⊖: once a feature is selected cannot be discarded

- SEQUENTIAL BACKWARD SELECTION (SBS): start with $F = Y$, and iteratively remove a non-relevant feature from F.
  Generalized SBS: remove the p least relevant features per iteration
  ⊖: more expensive (computationally)

- BIDIRECTIONAL GENERATION
- RANDOM GENERATION

MEASURES
- Information
- Distance
- Dependence
- Consistency
- Accuracy

(Information measures)

ENTROPY OF AN ATTRIBUTE: $E(Y_i) = -\sum_{v \in Y_i} P(v) \log_2 P(v)$

ENTROPY OF X AFTER OBSERVING Y

$$E(Z|Y_j) = \sum_{v \in Y_j} P(v) E(Z|v)$$

INFORMATION GAIN: $IG(Z|Y_j) = E(Z) - E(Z|Y_j)$

$\chi^2$ test (for categoric features)
ANOVA test (for numeric features)

(Distance measures → see clustering)

CONSISTENCY MEASURES: minimum set of features that separate classes as consistently as fully set can
INCONSISTENCY: 2 instances having the same feature values but ≠ classes

**Accuracy measures**: objective criteria, dependent on the chosen classifier; need to apply a classifier.

**MODELS** (SUPERVISED)

① **FILTER** MEASURING!

→ feature selection independent from the mining algorithm

→ rely on general characteristics of data
- information
- distance
- dependence
- consistency

→ fast and no bias towards any learning algorithm

② **WRAPPER** LEARNING ALGORITHM!

→ evaluated on a given classification / regression algorithm

→ use predictive accuracy or sensitivity as a goodness measure

→ better performance yet dependent on the learning algorithm

→ more computationally expensive

**EVALUATION**

. Direct / objective metrics
- Feature selection: matching scores
- Feature ranking: correlation or distance between the real ranks and the estimated rank

. Indirect / subjective

→ learning performance on selected features
- predictive accuracy
- goodness of resulting clusters

→ nr of features selected

→ efficiency

**OTHER FORMS OF DATA REDUCTION**

① **DOMAIN REDUCTION**: reduce data representation

* **BINNING / DISCRETIZATION** (of real-valued attributes): converting a continuous value into a set of ranges
* **CARDINALITY REDUCTION** (of nominal and ordinal attributes): combining two or more categories into one new category.

② **SAMPLE NUMEROSITY REDUCTION**: reduce the nr of instances according to their similarity

* **SIMPLE RANDOM SUBSAMPLE (SRS)**: randomly remove observations per class
* **BALANCED SAMPLE**: remove observations guaranteeing that the reduced data have a certain composition according to a predefined criteria
* **STRATIFIED SAMPLE**: observations are divided into mutually disjointed parts, called strata, and removal (SRS) is done at each stratum
* **DATA CLUSTERING**: clustering the data and use centroids instead of the actual data
* **DATA CONDENSATION**: obtain minimal data set for correctly classifying all original observations (naive sampling is not suitable to learn from noisy data)
* **DATA SQUASHING**: compress data in such a way that a statistical analysis carried out on the compressed or original data yields the same outcome.

**Spearman correlation**

$$rs = \rho_{rgx, rgy} = \frac{cov(rgx, rgy)}{\sigma_{rgx} \times \sigma_{rgy}}$$

$\rho$: coeficiente de correlação de Pearson

$cov(rgx, rgy)$: covariância das variáveis

$\sigma_{rgx}, \sigma_{rgy}$: desvios padrão das variáveis

**covariance**

$$cov(x,y) = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{n-1}$$

**variance**

$$var(x) = \sqrt{\frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n-1}}$$

**Pearson correlation**

$$r_{xy} = \frac{cov(x,y)}{\sqrt{var(x) \cdot var(y)}}$$

= -1: fully inversely correlated

= 1: fully correlated

$$\sigma = \left[\frac{1}{n-1} \sum (x_i - \bar{x})^2\right]^{0,5}$$

spearman = 1 / pearson = 1

**Spearman** entre x e y

Rank 1 - Rank 2

| x | y | Rank 1 | Rank 2 | D | $D^2$ |
|---|---|---|---|---|---|
| | | | | | |

↓

caso haja empates fazer $\frac{a1 + a2}{2}$ e dar esse valor aos 2

$$Spearman = 1 - \frac{6 \sum D^2}{n \times (n^2 - 1)}$$

↳ # linhas

$$\sigma_{y1} = \left(\frac{1}{3}\left(-0,575^2 + 0,525^2 + 0,525^2 + (-0,475)^2\right)\right)^{0,5} = \left(\frac{1,1075}{3}\right)^{0,5} = 0,60759$$

$$\sigma_{y4} = \left(\frac{1}{3}\left(0,25^2 + (-0,75)^2 + 1,25^2 + (-0,75)^2\right)\right)^{0,5} = \left(\frac{2,75}{3}\right)^{0,5} = 0,9574$$

Exercises

(1)

| y1 |
|---|
| -0,4 |
| 0.7 |
| 0.7 |
| 0.7 |
| -0,3 |

$\bar{y1} = 0,175$

$\bar{y2} = 0,05$

$\bar{y3} = -0,325$

$\bar{y4} = 1,75$

Spearman (y1, y3)

| y1 | y3 | R1 | R2 | D | $D^2$ |
|---|---|---|---|---|---|
| -0,4 | 0,1 | 1 | 4 | -3 | 9 |
| 0,7 | -0,3 | 3,5 | 2 | 1,5 | 2,25 |
| 0,7 | -0,1 | 3,5 | 3 | 0,5 | 0,25 |
| -0,3 | -1 | 2 | 1 | 1 | 1 |

$$1 - \frac{6 \times (9 + 2,25 + 0,25 + 1)}{4 \times (4^2 - 1)} = 1 - \frac{75}{60} = -0,25$$

1.1. y1 = aut

(1) Spearman

$$\rho_{y1,y2} = \frac{cov(y1, y4)}{\sigma_{y1}, \sigma_{y4}} = \frac{0,1583}{0,60759 \times}$$

$$cov(y1, y4) = \frac{(-0,575) \times 0,25 + 0,525 \times (-0,75) + 0,525 \times 1,25 + (-0,475) \times (-0,75)}{3}$$

$$= \frac{-0,14375 - 0,39375 + 0,65625 + 0,35625}{3} = \frac{0,475}{3} = 0,1583$$

(2) Pearson correlation

$$\rho_{y1,y3} = \frac{cov(y1, y3)}{\sqrt{var(y1) \cdot var(y3)}} = \frac{0,06}{\sqrt{0,70}}$$

$$cov(y1, y3) = \frac{(-0,4 - 0,175) \times (0,1 + 0,325) + (0,7 - 0,175) \times (-0,3 + 0,325) + (0,7 - 0,175) \times (-0,1 + 0,325) + (-0,3 - 0,175) \times (-1 + 0,325)}{3}$$

$$= \frac{(-0,575) \times 0,425 + 0,525 \times 0,025 + 0,525 \times 0,225 + (-0,475) \times (-0,675)}{3}$$

$$= \frac{-0,244375 + 0,013125 + 0,118125 + 0,320625}{3} = 0,0691(6)$$

$$var(y1) = \sqrt{\frac{(-0,575)^2 + 0,525^2 + 0,525^2 + (-0,475)^2}{3}} = 0,20253$$

$$var(y3) = \sqrt{\frac{0,425^2 + 0,025^2 + 0,225^2 + (-0,675)^2}{3}} = 0,478714$$

$$\rho_{y1,y2} = \frac{cov(y1, y2)}{\sqrt{var(y1) \times var(y2)}} = \frac{-0,06167}{\sqrt{0,20253 \times 0,81445}} = -0,15 < 0$$

↓ inversely correlated

$$cov(y1, y2) = -0,06167$$

$$var(y2) = \sqrt{\frac{0,0225 + 0,3025 + 0,5625 + 1,1025}{3}} = 0,81445$$

(3) ?

1.2. Y4: CLASS ATTRIBUTE  Y1, Y2, Y3: DOMAIN ATTRIBUTES

(1) LOW-VARIANCE RANKING CRITERIA

(2) GAIN-BASED RANKING

(3) SINGLE SPLIT

(4) TWO SPLIT THRESHOLD

G(Y2) =

1.8 2 intervalos pq boolean

(1) Y1: [-0,3; -0,4], [0,7]

Y2: [-0,7; -0,1], [0,4; 0,6]

Y3 [-1, -0,3] [-0,1; 0,1]

Y4 [1, 1] [2, 3]

Y3: < 1, 0, 1, 0 >

1.8

(2) $w = \frac{max - min}{k}$

k → nº grupos em que quero dividir

Binário → k = 2

$w = \frac{0,1 + 1}{2} = \frac{1,1}{2} = 0,55$

[-1, -1 + 0,55[ [-1 + 0,55, 0,1]

[-1, -0,45[ [-0,45; 0,1]

Y3: < 1 0 0 0 >

1.3. DISCRETIZATION

- EQUAL-FREQUENCY: ... nd the maximum values, so ... erdu, and divides the range in ... tervals, in such a way that every interval contains the equal nr of sorted values
- EQUAL-WIDTH: first finds the minimum and maximum values of every variable, $x_i$, and then divides this range into a nr, $m x_i$, of user-specified, equal-width intervals.

$w = \frac{max - min}{k}$  [min, min + w[ [min + w, min + w + w[

2.1. - Subset selection can reduce overfitting

- Lasso regression frequently eliminates some of the features
- Finding the true best subset takes exponential time
- Wrappers: feature selection methods that, given a classifier as a performance criteria, search in the space of subset of features for the minimal one that obtains the higher accuracy

2.2. FORWARD SUBSET SELECTION

(2) → it greedily adds the feature that most improves cross-validation accuracy

(4) → FS is faster than BS if few features are relevant to prediction.

3.1. HIGH-DIMENSIONAL DATA

- GLOBAL LEARNING models have high propensity to overfit the original data as they cannot flexibly discard uninformative or non-discriminative subspaces
- LOCAL LEARNING models have high propensity to underfit the original data neglect informative and/or discriminative subspaces
- Statistically significant differences between training and testing errors denote generalization difficulty

3.2. STRATEGIES TO REDUCE OVERFITTING IN DECISION TREES:

- Pruning
- enforce a minimum nr of samples in leaf nodes
- " a maximum depth for the tree

1.3

(4) Gaussian cut-off points

ver quartil para probabilidade = $\frac{1}{4}$  <1,000

BINARIZING → discretization

# CLUSTERING

**CLUSTER:** collection of data objects
- SIMILAR (or related) to the objects within the same group
- DISSIMILAR (or unrelated) to the objects in other groups

**CLUSTER ANALYSIS:** group data objects into clusters according to their (dis)similarity
→ Intracluster distances are minimized
→ Intercluster " are maximized

**GRAPH-BASED PROBLEM:**
- proximity between all data points defines a weighted graph
- nodes are the objects, edges capture their distances
- clustering = breaking the graph into connected components
- want to minimize the edge weight between clusters, maximize the edge weight within clusters

CLUSTERING ± K-MEANS

**UNSUPERVISED:** cluster observations without knowing their outcome variable

**SEMI-SUPERVISED:** cluster observations when:
- labels of some observations may be known
- pairs of observations are known to belong to the same cluster

**SUPERVISED:** - cluster observations when their class is considered:
- label added as an additional attribute
- cluster class-conditional observations

**DISTANCE METRICS**

* DATA DOMAIN
→ numeric and ordinal (Euclidean)
→ nominal (Hamming)
→ mix of non-iid attributes

* DATA STRUCTURE: tabular, time series, images, spatio-temporal data, events...

## APPROACH

* **PARTITIONING**: create partitions and iteratively update them
  K-means, K-modes, K-medoids
* **HIERARCHICAL**: create hierarchical decomposition of data points
  Diana, Agnes
* **DENSITY-BASED**: group points based on connectivity and density functions
  DBSCAN, DenClue
* **MODEL-BASED**: data are seen as a mixture of distributions
  EM
* **GRID-BASED** STING

---

## PARTITIONING ALGORITHMS

Given K:

1) Partition objects into K non-empty subsets
2) Compute seed points as the centroid of each subset
   CENTROID: is the center (mean/median/mode point)
3) Reassign each object to the cluster with the nearest seed
4) Go back to step 2, stop when assignment does not change

VARIANTS:

→ selection of the initial seeds
→ similarity distances
→ timing of centroid calculus

## K-MEANS

① Arbitrarily choose K object as initial cluster center
② Assign each object to most similar center
③ Update the cluster means
④ Reassign

→ each data point can only be assigned in one cluster

EFFICIENCY: $O(tkn)$
$n$ = # objects
$k$ = # clusters
$t$ = # iterations

## PROBLEMS

→ dependent on initialization
→ sensitive to outliers
→ " to noisy data
→ not suitable to discover clusters with non-convex shapes
→ need to specify K in advance

→ Differing sizes
→ " Density
→ Non-globular shapes

} solution → use many clusters

## FUZZY C-MEANS

↳ data points can be assigned to more than one cluster
↳ each point has a degree of membership to belong to each cluster

⊕: relevant when individuals belong to different groups
⊖: need to define the nr of clusters, membership cutoff value; sensitive to initial assignment of centroids

## K-MODES and K-MEDOIDS

1) Numeric data: centroid as the mean
2) Categoric data: " " " mode
   ↳ K-modes: frequency-based procedure to update modes of clusters
3) Mixed data: centroid combining mean and mode (K-PROTOTYPE)

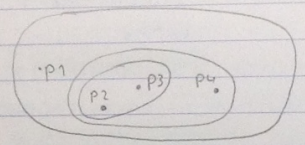
K-medoids: the most centrally located object in a cluster is the centroid

---

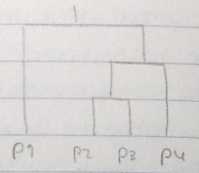
## HIERARCHICAL CLUSTERING

→ distance matrix between (non-clustered) data points and clusters
→ similarity matrix = - distance matrix
→ does not require fixed K

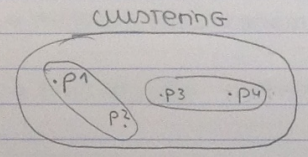
Traditional hierarchical clustering



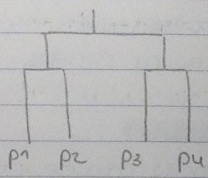

Traditional dendrogram




Non-traditional hierarchical clustering




Non-traditional dendrogram




**Nº of dendrograms** $= (2n-3)! / [(2^{(n-2)})(n-2)!]$

↳ nr of leafs

1) **Agglomerative** (bottom-up)
   - → initialize each point as its own cluster, find best pair
   - → iteratively merge clusters

2) **Divisive** (top-down)
   - → initialize all data points into one cluster
   - → large clusters are successively divided

**Distance between clusters**

1) Single link : smallest distance between points — MIN
2) Complete link : largest distance " " — MAX
3) Average link : average distance " "
4) Centroid link : distance between centroids

$$d(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{x_i \in C_i} \sum_{x_j \in C_j} d(x_i, x_j)$$

**Efficient graph-based algorithm:**

- → view all data points with links (distances) between them
- → finds maximal connected components in this graph
- → uses threshold distances at each level
- → could be agglomerative or divisive

MIN
- ⊕ : can handle non-elliptical shapes
- ⊖ : sensitive to noise and outliers

MAX
- ⊕ : less susceptible to noise and outliers
- ⊖ : tends to break large clusters ; biased towards globular clusters

---

**Density-based clustering** (DBSCAN)

- → clusters are defined as areas of higher density
- → separation occurs in sparse areas (data points seen as outliers)

---

**Model-based clustering**

- → Assume K probability distributions with parameters: $\theta_1, \dots, \theta_K$
- → given data X, compute $\theta_1, \dots, \theta_K$, such that $P(X \mid \theta_1, \dots, \theta_K)$ [likelihood] is maximized
- → Every point $x_i \in X$ can be generated by multiple distributions

**EM Algorithm**

- Initialize K cluster centers
- Iterate between 2 steps

Expectation step : assign points to clusters

$$P(x_i \in C_k) = w_k P(x_i \mid C_k) \Big/ \sum_j w_j P(x_i \mid C_j)$$

$$w_k = \frac{\sum P(x_i \in C_k)}{N}$$

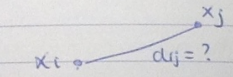
Maximization step : estimate model parameters

$$\mu_k = \frac{1}{m} \sum_{i=1}^{m} \frac{x_i P(x_i \in C_k)}{\sum_k P(x_i \in C_j)}$$

Distance function is a metric if:

- $d(x,y) \geq 0$
- $d(x,x) = 0$
- $d(x,y) = d(y,x)$
- $d(x,y) \leq d(x,z) + d(z,y)$

**Minkowski distance** (numeric data)




$$d(i,j) = \sqrt[q]{|x_{i1} - x_{j1}|^q + |x_{i2} - x_{j2}|^q + \dots + |x_{ip} - x_{jp}|^q}$$

$q = 2 \rightarrow$ Euclidean distance

$q = 1 \rightarrow$ Manhattan distance

$q = \infty \rightarrow$ Chebyshev distance

$$d_\infty(x,y) = \lim_{q \to \infty} \left( \sum_{i=1}^{n} |x_i - y_i|^q \right)^{1/q} =$$

$$= \max(|x_1 - y_1|, |x_2 - y_2|, \dots, |x_n - y_n|)$$

**Pearson correlation**

$$r_{xy} = \frac{cov(x,y)}{\sqrt{VAR(x) \cdot VAR(y)}}$$

$-1 \rightarrow$ fully inversely correlated

$1 \rightarrow$ totally correlated

**Covariance**: degree to which points are related

$$cov(x,y) = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{n-1}$$

**Variance**: difference degree from mean

$$VAR(x) = \sqrt{\frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n-1}}$$

**Hamming distance** (binary and categorical data)

↳ nr of different attribute values

K-Histograms

A B A
D D F
G H I

$\rightarrow$

| | | |
|---|---|---|
| A | | B |
| D | | F |
| G | H | I |

**Structure:**

Separation of clusters:

1) Exclusive
2) Non-exclusive

**Hard**: each object can only belong to single cluster

**Soft**: " " " belong to ≠ clusters

**Complete** vs **partial**

**Uniform** vs **weighted** attributes

**Evaluation:**

1) **External**: extent to which cluster labels match true labels

supervised: compare clusters against labels, prior or expert knowledge

ex entropy

2) **Internal**: goodness without external information

Silhouette: how well they are separated and how compact the clusters

unsupervised: criteria derived from data itself

ex sum of squared error

3) **Relative**: compare ≠ cluster analyses

ex: t-student tests

External measures

1) **Purity**

set of clusters — set of classes

$$purity(\Omega, C) = \frac{1}{n} \sum_k \max_j |\omega_k \cap c_j|$$

2) **Rand Index**

| | same class | ≠ class |
|---|---|---|
| same class | TP | FN |
| ≠ class | FP | TN |

$$RI = \frac{TP + TN}{All}$$

Precision = TP / (TP+FP)

Recall = TP / (TP+FN)

F-measure = $\frac{2 \times precision \times recall}{precision + recall}$

## Internal measures

1) **cohesion**: sum of the weight of all links within a cluster
↳ how closely related are points in a cluster

$$SSE = SSW = \sum_{k=1}^{K} \sum_{x_i \in C_k} d(x_i, C_k)^2$$

error in the distance to the nearest cluster

⊕ good to compare 2 clusters ⊖ when clusters are not well separated

2) **separation**: sum of the weights between nodes in the cluster and nodes outside the cluster
↳ how distinct or well-separated a cluster is from other clusters

$$SSB = BSS = \sum_{k} |C_k| d(C_k, \bar{x})^2$$

nr of cluster elements

ponto médio entre clusters usando o centro do cluster

3) **total error**

$$TSS = SSB + SSE$$

$$TSS = \sum_i d(x_i, \bar{x})^2$$

4) **silhouette coefficient**

$$s = \begin{cases} 1 - \frac{a}{b} & \text{if } a < b \\ 0 & \text{if } a = b \\ \frac{b}{a} - 1 & \text{if } a > b \end{cases}$$

closer to 1 ↓ better

a: average distance of $x_i$ to the points in its cluster
b: " " " " to points in another cluster

---

centroids don't reajust well
↓ solution
- multiple runs
- select more K initial centroids
- postprocessing

## Exercises

1.

| | Y1 | Y2 | class | |
|---|---|---|---|---|
| A1 | 2 | 10 | A | |
| A2 | 2 | 5 | A | |
| A3 | 8 | 4 | B | – |
| A4 | 5 | 8 | A | – |
| A5 | 7 | 5 | B | – |
| A6 | 6 | 4 | B | – |
| A7 | 1 | 2 | B | |
| A8 | 4 | 9 | A | – |

1.1. Initial clustering seeds: A1, A4 and A7

K1 = (2,10) – A1
K2 = (5,8) – A4
K3 = (1,2) – A7

→ calcular distâncias do cada ponto a cada um dos centróides (K1, K2, K3)

$d(A2, K1) = \sqrt{(2-2)^2 + (5-10)^2} = \sqrt{5^2} = 5$
$d(A3, K1) = \sqrt{(8-2)^2 + (4-10)^2} = \sqrt{36+36} = \sqrt{72} \simeq 8,485$
$d(A5, K1) = \sqrt{(7-2)^2 + (5-10)^2} = \sqrt{5^2+5^2} = \sqrt{50} \simeq 7,071$
$d(A6, K1) = \sqrt{(6-2)^2 + (4-10)^2} = \sqrt{4^2+(-6)^2} = \sqrt{52} \simeq 7,21$
$d(A8, K1) = \sqrt{(4-2)^2 + (9-10)^2} = \sqrt{2^2+(-1)^2} = \sqrt{5} \simeq 2,236$

$d(A2, K2) = \sqrt{(2-5)^2 + (5-8)^2} = \sqrt{9+9} = \sqrt{18} = 4,24$
$d(A3, K2) = \sqrt{(8-5)^2 + (4-8)^2} = \sqrt{9+36} = \sqrt{45} = 6,708$
$d(A5, K2) = \sqrt{(7-5)^2 + (5-8)^2} = \sqrt{4+9} = \sqrt{13} \simeq 3,606$
$d(A6, K2) = \sqrt{(6-5)^2 + (4-8)^2} = \sqrt{1+16} \quad \sqrt{17} \simeq 4,123$
$d(A8, K2) = \sqrt{(4-5)^2 + (9-8)^2} = \sqrt{1+1} = \sqrt{2} \simeq 1,4142$

$d(A2, K3) = \sqrt{(2-1)^2 + (5-2)^2} = \sqrt{1+9} = \sqrt{10} \simeq 3,162$
$d(A3, K3) = \sqrt{(8-1)^2 + (4-2)^2} = \sqrt{49+4} = \sqrt{53} \simeq 7,28$
$d(A5, K3) = \sqrt{(7-1)^2 + (5-2)^2} = \sqrt{36+3^2} = \sqrt{45} \simeq 6,708$
$d(A6, K3) = \sqrt{(6-1)^2 + (4-2)^2} = \sqrt{25+4} = \sqrt{29} \simeq 5,385$
$d(A8, K3) = \sqrt{(4-1)^2 + (9-2)^2} = \sqrt{9+49} = \sqrt{58} = 7,6157$

(A1) K1 → –
(A4) K2 → A3, A5, A6, A8, A4
(A7) K3 → A2, A7

(mean, mean)

K-median | K-mean | K-medoids

K1 | K1 | K1

$K2 = \left(\frac{8+4}{2}, \frac{9+4}{2}\right)$ $K2 = (6,6)$ (ver o main untral)

$= (6, 6.5)$ $K3 = (1.5; 7)$

$K3 = \left(\frac{2+1}{2}, \frac{5+2}{2}\right)$

$= (1,5 ; 3,5)$ (1)

1.2. Agglomerative clustering

(2)

1.3. $C1 = \{A1, A4, A8\}$ $C2 = \{A3, A5, A6\}$ $C3 = \{A2, A7\}$

Purity $(C3) = \frac{1}{2}$

Purity of clustering $= \frac{1}{8} \times (3+3+1) = \frac{7}{8}$

Rand Index: $\frac{TP + TN}{AU} = \frac{6+15}{6+15+6+1}$

TP: vem em cada cluster e ver a maioria de classe, foi combinado.
TN: o que está bem separados
FP: está junto e não devia
FN: está separado e não devia

$TP = C_3^2 + C_3^2 = 3+3 = 6$
$TN = 9+6 = 15$
vem do cluster C3 e os outros
$FN = 6$
$FP = 1$

(2)

2.

| | Y1 | Y2 | cluster |
|---|---|---|---|
| x1 | 2 | 2 | C1 |
| x2 | 2 | 4 | C1 |
| x3 | 4 | 3 | C2 |
| x4 | 4 | 5 | C2 |

2.1. b>a

Silhouette index $= 1 - \frac{a}{b} = 1 - \frac{2}{\sqrt{5}} = 0,1$

Euclidean

$d(x2, x1) = \sqrt{(2-2)^2 + (4-2)^2} = 2$ $a = \frac{2}{1} = 2$

$d(x2, x3) = \sqrt{(2-4)^2 + (4-3)^2} = \sqrt{4+1} = \sqrt{5}$ $b = \frac{\sqrt{5}+\sqrt{5}}{2} = \sqrt{5}$
$d(x2, x4) = \sqrt{(2-4)^2 + (4-5)^2} = \sqrt{5}$

Chebyshev

$d(x2, x1) = \max(|2-4|, |2-2|) = 2$ $a = 2$
$d(x2, x3) = \max(|2-4|, |4-3|) = 2$ $b = \frac{4}{2} = 2$
$d(x2, x4) = \max(|2-4|, |4-5|) = 2$

$a = b \rightarrow si = 0$

(x3)

$si(x3) = 1 - \frac{2}{3} = \frac{1}{3}$

$d(x3, x1) = |4-2| + |3-2| = 2+1 = 3$ $b = \frac{3+3}{2} = \frac{6}{2} = 3$
$d(x3, x2) = |4-2| + |3-4| = 2+1 = 3$
$d(x3, x4) = |4-4| + |3-5| = 0+2 = 2$ $\rightarrow a = 2$

(x4)

$si(x4) = 1 - \frac{2}{4} = \frac{1}{2}$

$d(x4, x1) = |4-2| + |5-2| = 2+3 = 5$ $b = \frac{3+5}{2} = \frac{8}{2} = 4$
$d(x4, x2) = |4-2| + |5-4| = 2+1 = 3$
$d(x4, x3) = 2$

Average $= \frac{\frac{1}{3} + \frac{1}{2}}{2} = \frac{\frac{2}{6} + \frac{3}{6}}{2} = \frac{\frac{5}{6}}{2} = \frac{5}{12} = 0,41(6)$

$C_1 = \left(\frac{2+2}{2}, \frac{2+4}{2}\right) = (2,3)$

$C_2 = \left(\frac{4+4}{2}, \frac{3+5}{2}\right) = (4,4)$

2.2. Euclidean distance

cohesion error $SSE = (2-2)^2 + (2-3)^2 + (2-2)^2 + (4-3)^2 + (4-4)^2 + (3-4)^2 + (4-4)^2 + (5-4)^2 =$
$= 0 + 1 + 0 + 1 + 0 + 1 + 0 + 1 = 4$

separation error $BSS = 2 \cdot (\sqrt{1,25})^2 + 2 \cdot (\sqrt{1,25})^2 = 5$

$d(C_1, PM) = \sqrt{(2-3)^2 + (3-3,5)^2} = \sqrt{1 + 0,5^2} = \sqrt{1,25}$

$d(C_2, PM) = \sqrt{(4-3)^2 + (4-3,5)^2} = \sqrt{1 + 0,5^2} = \sqrt{1,25}$

Ponto medio clusters $= \left(\frac{2+4}{2}, \frac{3+4}{2}\right) = (3; 3,5)$

$TSS = SSE + BSS = 9$

3.

| | Y1 | Y2 |
|---|---|---|
| A | 2 | 7 |
| B | 3 | 6 |
| C | 5 | 2 |
| D | 4 | 1 |

## PRINCIPAL COMPONENT ANALYSIS (PCA)

### Covariance matrix

$A = C^{n \times m} = (c_{ij} \mid i,j = 1..m)$ where $c_{ij} = cov(y_i, y_j)$

**Eigenvectors**: vectors $x$ having same direction as $Ax$
↳ define the linear composition of attributes

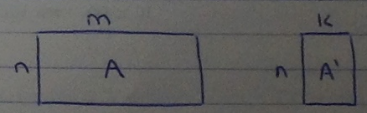
$Ax = \lambda x$
↳ **Eigenvalue** of A

$Ax = \lambda x \Leftrightarrow (A - \lambda I)x = 0$
↓ calculate $x$ and $\lambda$
→ determine roots to $\det(A - \lambda I) = 0$, roots are eigenvalues $\lambda$
→ solve $(A - \lambda I)x = 0$ for each $\lambda$ to obtain eigenvectors $x$

### Dimensionality Reduction

PCA

↳ map data with m attributes into k-derived attributes




"Residual" variation: info in A not retained in A'

### Linear Transformations

1. find the axis that shows the greatest variation
2. project all points into the axis

↓

- move origin to the center of the dataset
- find the eigenvectors and eigenvalues of the data covariance matrix ⇒ these define the new space

$1^{st}$ singular vector: direction of maximal variance
$\lambda_1$: how much of data variance is explained by $1^{st}$ vector

## PRINCIPAL COMPONENT ANALYSIS

→ project data points into orthogonal axes that best explain data

→ is SVD done on centered data
singular vector / value = eigenvector / value

→ eigenvector and eigenvalues computed on covariance matrix

→ projects data along the directions where the data varies most
↳ these directions determined by eigenvectors corresponding to the largest eigenvalues
↓
PRINCIPAL COMPONENTS : linear combinations of original features

→ it preserves as much information as possible by minimizing the reconstruction error

→ singular values : used to estimate how many components to keep

RULE OF THUMB : keep enough to explain 85% of the variation

$$\frac{\sum_{j=1}^{k} \lambda_j^2}{\sum_{j=1}^{n} \lambda_j^2} \simeq 0{,}85 \qquad k = m \Rightarrow 100\%$$

PC1 : eigenvalue with the largest absolute value
↓
first (direction along which there is greatest variation)

PC2 : direction with maximum variation left in data
↓
second orthogonal to previous components

COMPONENTS
- linear combinations of the original variables
- uncorrelated with each other
- capture as much of the original variance as possible

1) subtract mean of each attribute (centering data)
2) compute the covariance matrix $m \times m$
3) compute eigenvalues of $C$ : $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_m$ and eigenvectors of $C$ : $e_1, e_2, \dots, e_m$
4) keep the largest $k$ eigenvalues : dimensionality reduction ($k \leq m$) and construct the transformed space
5) transform the dataset $A \to A'$

Transformed Feature vector
$k = 2 \rightarrow$ Feature vector $= \begin{bmatrix} e_1 & e_2 \end{bmatrix}$

Transformed Data = Row Featurevector × Row Data Adjust

Linear Discriminant Analysis (LDA) : data transformation guaranteeing class separation

pick new dimension that
- maximize separation between means of projected classes
- minimize variance for the observations within each class

↓

Eigenvectors based on between-class and within-class covariance matrices

## RECOVERING ORIGINAL DATA

Data Recovered = (Featurevector × Transformed Data) + original mean
= Data Adjust + original mean

RECONSTRUCTION ERROR : $\| x - \hat{x} \|$

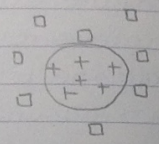
$$\text{error} = \frac{1}{2} \sum_{i=k+1}^{m} \lambda_i$$

## Nonlinear PCA using kernels (Traditional applies linear transformation)

$\phi : x \rightarrow \phi(x)$

PCA rewritten in terms of dot product

$K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$

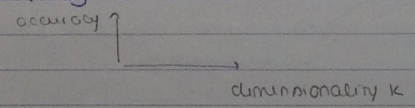


## Evaluation

- Direct: similarity scores suitable for artificial data or data with prior knowledge

- Indirect:
  - → learning performance on the transformed space
    - predictive accuracy
    - goodness of resulting clusters

    Before-and-after performance
  - → reconstruction error
  - → learning curves




## Regression

- Descriptive / unsupervised setting: describe relation between a set of (explanatory) attributes and a target real-valued attribute
- Predictive / supervised setting: learn a mapping to estimate the outcome of new observations. (real-valued outcome)

Given a dataset and a real-valued attribute of interest

↓

Regression models the relationship between predictors and a target real-valued variable

Explanatory modeling: understand how predictors explain the target variable

Predictive modeling: estimate target values from predictor values.

↓

fit the data by optimizing predictive accuracy

REGRESSION

**Correlation**: degree to which 2 attributes are related
- the sign: nature of association (>0 direct; <0 inverse)
- the absolute value of r: strength of association

1 or −1: perfect correlation  0: no relation

## Pearson correlation coefficient

→ only suitable for numeric attributes

→ able to handle scales and shifts

$$r = \frac{\sum y_1 y_2 - \frac{\sum y_1 \sum y_2}{n}}{\sqrt{\left(\sum y_1^2 - \frac{(\sum y_1)^2}{n}\right) \cdot \left(\sum y_2^2 - \frac{(\sum y_2)^2}{n}\right)}}$$

## Spearman rank

↳ when attributes are either numeric and/or ordinal

1) rank the values of $y_1$ and $y_2$
2) compute the value of $d_i$ for each pair of observations by subtracting the rank of $y_{2i}$ from the rank of $y_{1i}$
3) Apply the formula

$$r_s = 1 - \frac{6 \sum d_i^2}{n(n^2-1)}$$

↓

magnitude and nature of association

How to approximate the observed correlation?
(Best-fit line)

↓

minimize residuals

Regression lines
- Global (↑ single/whole function)
  - Linear
  - Polynomial
  - Others
- Local (↑ piecewise function)

## Linear Regression

↳ least squares method: minimizes squared deviations between line and data points

$$\hat{z} = \bar{z} + b(y - \bar{y}) = \beta_0 + \beta_1 y$$

$b$ ↓ slope; $\bar{z} - b\bar{y}$ ↘ vertical axis intercept

$$b = \frac{\sum yz - \frac{\sum y \sum z}{n}}{\sum y^2 - [(\sum y^2)/n]}$$

Lazy regression – KNN

## Multivariate linear regression

: more than one attribute

$$z = \beta_0 + \beta_1 y_1 + \dots + \beta_m y_m + \varepsilon$$

↓

numeric predictor attributes $\{y_1, \dots, y_m\}$

**Partial regression coefficients**: $\beta_i \equiv$ effect (on the mean response) of increasing $i^{th}$ predictor attribute by 1 unit

**Train**: estimate parameters

$$\text{data } A = \begin{bmatrix} 1 & a_{11} & \dots & a_{1m} \\ \dots & \dots & \dots & \dots \\ 1 & a_{n1} & \dots & a_{nm} \end{bmatrix} \quad \text{outcome } z = \begin{bmatrix} z_1 \\ \dots \\ z_n \end{bmatrix}$$

$$\hat{\beta} = (A^T A)^{-1} A^T z \quad \text{where} \quad \hat{\beta} = \begin{bmatrix} \hat{\beta}_0 \\ \dots \\ \hat{\beta}_n \end{bmatrix}$$

**Test**:

$$x_{new} = [a_{new1} \dots a_{new m}]$$

$$\hat{z} = \hat{\beta}_0 + \hat{\beta}_1 a_{new1} + \dots + \hat{\beta}_m a_{new m}$$

**Ordinary least squares estimation**:

$$\min SSE = \sum (z - \hat{z})^2$$

**LASSO estimation**:

$$\min SSE = \sum_{i=1}^{n} (z - \hat{z})^2 + \sum_{j=1}^{p} |\beta_j|$$

**Ridge regression estimation**:

$$\min SSE = \sum_{i=1}^{n} (z - \hat{z})^2 + \sum_{j=1}^{p} |\beta_j^2|$$

## NON-LINEAR REGRESSION

1) Nonlinear regression problems can be moved to a linear domain by a suitable transformation of the model formulation

2) nonlinear data can be split up into segments and linear regression performed per segment

$$f(y,w) = \beta_0 + w_1 y + w_2 y^2 + \dots + w_N y^n = \sum_{j=0}^{N} w_j y^j$$

$y = [y_1 \dots y_m]$ and $w_i = [w_{i1} \dots w_{im}]^T$

## NON-PARAMETRIC REGRESSION

- k-nearest-neighbors
- decision tree regression
- naive: discretize target variable and apply traditional classifiers

## LOCAL REGRESSION

→ square smoothing
→ linear smoothing
→ kernel smoothing

small k (KNN) and h (smoothers) and high depth (decision trees):
→ single instances matter
→ bias is small
→ variance is large

k↑ h↑ depth↑ → average over more instances
↓
variance↓
bias ↑

cross-validation to finetune k, depth or h

## EVALUATION

ON TESTING DATA — cross-fold validation

- Mean absolute error $MAE(\hat{z},z) = \frac{1}{n}\sum_{i=1}^{n} |z_i - \hat{z}_i|$
- Mean square error $MSE(\hat{z},z) = \frac{1}{n}\sum_{i=1}^{n} (z_i - \hat{z}_i)^2$
- Root mean square error $RMSE(\hat{z},z) = \sqrt{\frac{1}{n}\sum^{n} (z_i - \hat{z}_i)^2}$
- Mean absolute percentage error
  $MAPE(\hat{z},z) = \frac{1}{n}\sum^{n} \left| \frac{z_i - \hat{z}_i}{z_i} \right|$

ON TRAINING DATA

- Coefficient of determination: measures the goodness of the fitting
  $R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST}$
- Sum of squares total: $SST = \sum_{i=1}^{n} (z_i - \bar{z})^2$
- Sum of squares regression: $SSR = \sum_{i=1}^{n} (\hat{z}_i - \bar{\hat{z}})^2$
- Sum of squares error: $SSE = \sum_{i=1}^{n} (z_i - \hat{z}_i)^2$

$SST = SSE + SSR$

* Analysis of residuals (errors)
↓
$\boxed{\hat{z} - z}$

Similarity $= 1 - \frac{d}{d_{max}}$

→

## DISTRIBUTION OF DISTANCE WEIGHTS

1º soma das similaridades dos vizinhos a considerar desse ponto

2º $\frac{\text{similaridade}}{\text{soma (total)}}$

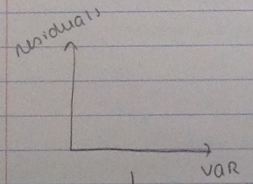
$$w_{12} = \frac{s_{12}}{\text{soma singularidades}}$$

**QUANTITY FOR x1** when considering k = 2 ...

$$\hat{z}_1 = z_5 \times w_{15} + z_6 \times w_{16}$$

PQ $x_5$ e $x_6$ os 2 pontos + próximos de $x_1$

$$w_{15} = \frac{s_{15}}{s_{15} + s_{16}} \qquad w_{16} = \frac{s_{16}}{s_{15} + s_{16}}$$




- se o modelo está biased
- se erros estão muito altos

↓

tem tendencia → MAU

↓

para alguns valores o estimador tem maus resultados

↓

substima sempre o valor real

## BICLUSTERING

**BICLUSTER** ($B = (I, J)$) : $n \times m$ submatrix of A, where $I = (x_{i1}, \ldots, x_{in}) \subset X$ is a subset of rows and $J = (y_{j1}, \ldots, y_{jm}) \subset Y$ is a subset of columns.

**BICLUSTERING** : identify a set of biclusters $B = \{B_1, \ldots, B_s\}$ such that each bicluster $B_k = (I_k, J_k)$ satisfies specific criteria of: → homogeneity
→ statistical significance.

USED TO:

1) Find patterns on real-valued data
2) Find less-trivial patterns with non-constant homogeneity
3) Describe or classify data using discriminative patterns
4) Dimensionality reduction by reducing data into a set of biclusters

GLOBAL: clustering $\xrightarrow{\text{PROBLEM}}$ HIGH-DIMENSIONAL DATA
↓ SOLUTION
LOCAL: **BICLUSTERING**

→ group observations correlated on a subset of all attributes
→ non-exhaustive and overlapping groups of observations

**HOMOGENEITY**: patterns of interest

**STATISTICAL SIGNIFICANCE**: whether the probability of a bicluster to occur deviates from expectations

- non-significant bicluster discovered: $\boxed{FP}$
- significant bicluster not discovered: $\boxed{FN}$

HOMOGENEITY – it determines
- structure (positioning)
- coherence (correlation) variance of the values in a bicluster
- quality (noise tolerance)

**COHERENCE** . variance of the values in a bicluster

**1) CONSTANT MODEL**

overall: $a_{ij} = K + \eta_{ij}$

on rows: $a_{ij} = K_i + \eta_{ij}$

on columns: $a_{ij} = K_j + \eta_{ij}$

↳ noise factor

**2) ADDITIVE MODEL**

value of column $x_j$

on rows: $a_{ij} = c_j + \gamma_i + \eta_{ij}$ ↳ adjustment for row $x_i$

on columns: $a_{ij} = c_i + \gamma_j + \eta_{ij}$

↓ value of row $x_i$ ↳ adjustment for column $\gamma_j$

Bicluster **PATTERN $\varphi_B$** : expected values in the absence of adjustments $\gamma_i$ and noise factors $\eta_{ij}$

$\varphi_B =$ 2 1 4 2

| | $y_1$ | $y_2$ | $y_3$ | $y_4$ | |
|---|---|---|---|---|---|
| $x_1$ | 3 | 2 | 5 | 3 | $\varphi_B + 1$ |
| $x_2$ | 2 | 1 | 4 | 2 | $\varphi_B + 0$ |

**3) MULTIPLICATIVE MODEL**

on rows: $a_{ij} = c_j \gamma_i + \eta_{ij}$

on columns: $a_{ij} = c_i \gamma_j + \eta_{ij}$

$\varphi_B =$ 1 4 2

| | $y_1$ | $y_2$ | $y_3$ | |
|---|---|---|---|---|
| $x_1$ | 1 | 4 | 2 | $\varphi_B \times 1$ |
| $x_2$ | 3 | 12 | 6 | $\varphi_B \times 3$ |

**4) ORDER-PRESERVING MODEL**

(I,J) where the values on each row in I across the J columns are ordered according the same permutation $\pi$

| | $y_2$ | $y_4$ | $y_5$ |
|---|---|---|---|
| $x_1$ | 2 | 3 | 4 |
| $x_2$ | 1 | 2 | 2 |
| $x_3$ | 2 | 1 | 4 |

$y_2 \leq y_5 \leq y_4$

**MERIT FUNCTIONS:**

1) variance of values ⇒ constant overall
2) pearson correlation ⇒ additive / multiplicative
3) cosine measure ⇒ order-preserving
4) mean square residue ⇒ additive

**COHERENCE STRENGTH**

$\bar{A}$: amplitude of the range of values in a matrix A

coherence strength: range $\delta \in [0, \bar{A}]$, such that $a_{ij} = c_j + \gamma_i + \eta_{ij}$ (or other) and $\eta_{ij} \in [-\delta/2, \delta/2]$

> coherence strength
- + tolerance to noise
- larger biclusters

**QUALITY** . type and amount of accomodated noise $[\eta_{ij} - \delta/2, \eta_{ij} + \delta/2]$

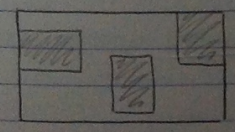
**STRUCTURE:**

→ # biclusters
→ size and shape
→ positioning constraints

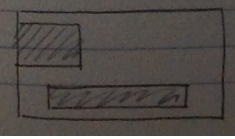
**POSITIONING CONSTRAINTS**

**1) Exclusive**

- on columns: não há 2 colunas que pertençam a 2 biclusters

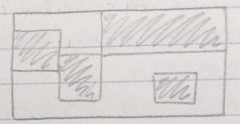


- on rows: não há nenhuma row que pertença a 2 biclusters

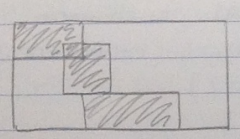


2) Exhaustive

- on columns : vai da esquerda à direita



- on rows : vai de baixo a cima



4) Non-overlapping

- on columns : consegue-se separar colunas
- on rows : consegue-se separar rows
- Biclusters : biclusters separados

5) Hierarchical : uns dentro de outros

6) Flexible : arbitrarily positioned biclusters

Statistical significance

Is a bicluster unexpected?

↓

- randomize data several times and compute the ratio of datasets where we are able to find a similar bicluster
- approximate data distributions and perform a statistical test

Approaches

- number of biclusters at a time
  - discover one bicluster at a time and then mask it
  - " a set of biclusters at a time
- optimally:
  - Exhaustive searches
  - Approximate searches
    - greedy iterative add/remove rows and columns
    - row and column clustering combination
    - divide and conquer
    - distribution parameter reinstatement

Evaluation

- Synthetic data : knowledge of true/hidden biclusters
  Objective metrics: accuracy, precision, completeness
- Real data : no knowledge of true biclusters
  Subjective metrics : domain-driven relevance scores

Synthetic data

→ clustering metrics on one dimension (entropy, recall, precision)

→ Jaccard

$$MS(B,H) = \frac{1}{|B|} \sum_{(I_1,J_1) \in B} \max_{(I_2,J_2) \in H} \left| \frac{I_1 \cap I_2}{I_1 \cup I_2} \right|$$

→ FABIA consensus

$$FC(B,H) = \cdots$$

Real data

→ statistical significance : unexpected occurrence probability

→ domain relevance : unexpected probability of participating in studied process

## DATA TYPES

- Time series data
- Network data
- 3-dimensional data
- N- " "
- Heterogeneous data