APUNTS:

Mineria de dades

Data Mining notes

MIDA

Mariona Farré

UPC - EPSEVG

Curs: 2023-24

Tema 1: Introducció a la Mineria de dades **4**
Raw data: 5
Tema 2.1: Caracterització i preparació de les dades **7**
Preparació de les dades: **7**
Forma Tabular: 7
Entendre les dades: **8**
Numerals: 8
Categorical: 9
Anàlisi de columnes: **9**
Uni-variate: 9
Boxplot: 9
Histograma: 9
Estudi bi-variate: 10
Dues quantitatives: 10
Qualitativa i Quantitativa: 10
Dues qualitatives: 10
Processament de dades: **10**
Data Cleaning: 10
Missing data: 10
Trobar errors: 11
Finding outliers: 11
1. Gràfic: 12
2. Statistical-based: 12
3. Distance-based i Density-based: 12
Tema 2.2: Transformació de Dades **13**
Value transformation: **13**
Homogeneïtzació de les dades: 13
Categorical a Numeral: 13
● Ordinal (Label) encoding: 13
● One-Hot encoding: 13
● Mean encoding: 13
Numerical to Qualitative: 13
Reescalar les dades: 14
Canviar la distribució: 14
Effect of log-normalization: 14
Reemplaçar per quantiles: 14
Treball en les columnes: **14**
Reduir el número de columnes: Treure variables 14
● Feature Selection: 14
● Feature Extraction: 16
Enriquir el dataset: Ajuntar taules 17
Enriquir el dataset: Crear noves variables 17
Treball en les files: **17**

Reduir el nombre de files: 17
Sampling: 17
Incrementar el nombre de files: 18
Datasets no balancejats: 18
Tema 3: Mètodes supervisats: Intro i avaluació 19
Avaluació de classificadors: 19
Introducció i mesures: 19
Cross-validation: 20
K-fold Cross-validation: 21
Leave-one-out (LOO): 21
Bootstrapping: 21
Confidence intervals: 21
Multi-class classification: 22
One vs One(OVO) 22
One vs All (OVA) 22
Data preparation: 23
Generalitzacions al crear classificadors: 23
Discriminative method: 23
Generative method: 23
Tema 4.1: knn i Naive Bayes 24
Nearest Neighbor: 24
K-NN: 24
Mesures de distància: 24
Algoritme: 24
Distància amb pes: 25
Naïve Bayes: 26
teorema de Bayes: 26
Tema 4.2: Decision Trees 28
Top-Down: Induction de Decision Trees 28
Teoria de la informació: 29
Resposta binària: 29
Entropia (Entropy): 29
Overfitting: 30
Variables categòriques: 31
Atributs amb valors numèrics: 31
Arbre de decisió per regressió: 31
Tema 5.1 Boosting 32
Esquemes de votació: 32
Stacking: 32
Weighted majority: 32
Bagging and Random Forest: 33
Bagging: 33
Random Forests: 33
Boosting: 33

Adaboost algorithm: 34
Boosting vs Bagging: 34
Tema 5.2 Support Vector Machines **35**
Large-margin linear classifier: **35**
Linear Separable: 35
Trobar el límit de decisió: 35
Nonlinear Separable Datasets: Soft-Margin SVM 37
Non-linear classifier: Kernel trick: **37**
Transformar dades en una dimensió més: 37
Tipus de Kernel: **38**
Tema 6: Clustering **39**
Clustering: **39**
Distàncies i Mesures: **39**
Mesura de similitud: 39
Algorismes: **39**
Cluster K-means: 40
Evaluar K-means clusters: 40
Preprocessament i Postprocessament: 41
Limitacions: 41
Hierarchical clustering: 41
Agglomerative: 42
MST: Divisive Hierarchical Clustering: 44
Spectral/Graph-based Clustering: 44
DBScan: Algorismes amb Estimadors Densitat: 44
Evolució dels clusters: **45**
Tema 7: Regles d'Associació **47**
Aplicacions: 47
Regla d’associació: 47
2 Step approach: 48
Força bruta approach: 48
Reduir nombre de Candidats: 49
Apriori principle: 49

Tema 1: Introducció a la Mineria de dades

pq: ordinadors, internet,compres digitals→ tot moment es genera dades(qui,quan,que,com,que passa…)
important per poder extreure info
mineria de dades els algorismes poden ser +/- eficients - max cost lineal o logarítmic

S'utiliza a totes les empreses que poden tenir un rastre de la informació
Si no s'utilitza mineria de dades, les empreses només analitzen 5-10% de les dades que poden ser analitzades, però mina intenta tractarles totes les dades possibles
empreses volen un resum, extreure info dels usuaris i info de l'empresa
les dades no es lo important-> la info->el coneixement de les dades-> sabiduria per entendre

Medical Data by Dr. Tsumoto, Tokyo Med. & Dent. Univ., 38 atributs

```
...
10, M, 0, 10, 10, 0, 0, 0, SUBACUTE, 37, 2, 1, 0,15,-,-, 6000, 2, 0, abnormal, abnormal,-, 2852, 2148,
712, 97, 49, F,-,multiple,,2137, negative, n, n, ABSCESS,VIRUS
12, M, 0, 5, 5, 0, 0, 0, ACUTE, 38.5, 2, 1, 0,15, -,-, 10700,4,0,normal, abnormal, +, 1080, 680, 400, 71,
59, F,-,ABPC+CZX,, 70, negative, n, n, n, BACTERIA, BACTERIA
15, M, 0, 3, 2, 3, 0, 0, ACUTE, 39.3, 3, 1, 0,15, -,-, 6000, 0,0, normal, abnormal, +, 1124, 622, 502, 47,
63, F, -,FMOX+AMK, , 48, negative, n, n, n, BACTE(E), BACTERIA
16, M, 0, 32, 32, 0, 0, 0, SUBACUTE, 38, 2, 0,    0, 15, -, +, 12600, 4, 0,abnormal, abnormal, +, 41, 39,
2, 44, 57, F, -, ABPC+CZX, ?, ? ,negative, ?, n, n, ABSCESS,    VIRUS
...
```

Attribut Numéric — Attribut categoric — valors perduts — Etiqueta de classe

No es pot processar al veure la informació crua i sense analitzar

```
IF  a9=ACUTE        AND   a33=negative AND      a2=m AND a10<30
THEN        BACTERIA    [87,5%]
```

Precisió en la predicció

Les prediccions dels diferents resultats a traves de les dades creant regles i donant un percentatge de predicció de la norma

Mineria dades: un PROCÉS que per extracció automàtica de coneixement ocult **no obvi** a partir de conjunts de dades de gran volum

- processament no lineal ciclic fins arribar resultat
- Trobar patrons/models al conjunt de dades
- Descobrir info desconeguda oculta entre les dades

Son les dades q generen les tesis dels possibles algorismes

- Estadística: construcció dels models algoritmics
- bds: gestió grans quantitats dades
- Aprenentatge automàtic: en algorismes
- Visualització de dades: més facil entendible

Acme Investors Incorporated

Customer ID	Account Type	Margin Account	Transaction Method	Trades/ Month	Sex	Age	Favorite Recreation	Annual Income
1005	Joint	No	Online	12.5	F	30–39	Tennis	40–59K
1013	Custodial	No	Broker	0.5	F	50–59	Skiing	80–99K
1245	Joint	No	Online	3.6	M	20–29	Golf	20–39K
2110	Individual	Yes	Broker	22.3	M	30–39	Fishing	40–59K
1001	Individual	Yes	Online	5.0	M	40–49	Golf	60–79K

Clustering:	• Puc desenvolupar una caracterització general / Perfil de diferents tipus d'inversors?
Classificació	• Quines característiques distingeixen entre els inversors i el Broker Online?
Descobriment de patrons	• Puc descobrir quan l'ocurrencia d'uns events estan relacionats amb d'altres?

Classificació
- trobar la descripció de diverses classes predefinides i classificar un element de dades en un d'ells.

Regressió
- assigna una dada a una variable de predicció de valor real

Clustering
- la identificació d'un conjunt finit de categories o grups per descriure les dades (p.e. la personalització)

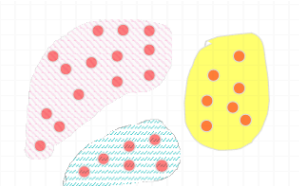
Sumarització
- la recerca d'una descripció compacta per a un subconjunt de les dades

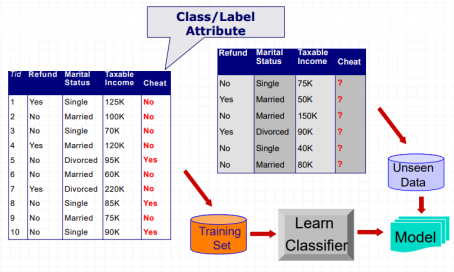
Modelització de dependències
- la recerca d'un model que descriu les dependències significatives entre les variables.

Detecció de canvis i desviacions
- descobrir canvis significatius en les dades

tipus estudi

- **Clustering:** exploració de dades per entendre contingut donat dades, buscar dintre d'aquestes **similituds** entre dades
  Dades dins d'un mateix cluster son similars
  Les dades en diferents clusters son diferents entre elles
  → Resultat: descriptiu agrupacions de les dades (generar clusters de informació)





- **Classificació**: donat un training set
  - les dades amb certs atributs
  - Uns dels atributs de la classe podria ser **preedit**

  Trobar un Model d'un atribut com una funció de els valors dels altres atributs
  →Resultat: noves característiques es poden assignar a classe amb molta accuracy
  Test set per verificar verificació del model/ training set per entrenar i validar les dades

- **Descobriments de patrons:**
  Donat un set de dades amb un num de items de la col·lecció
  → Produeixen **normes de dependencia** que prediran la ocurrència d'un item basat en la recurrencia dels altres items

TID	Items
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

Rules Discovered
{Diaper, Milk} → {Beer}
{Milk} → {Coke}

- **Regressió :** assigna una dada a una variable de predicció de valor real

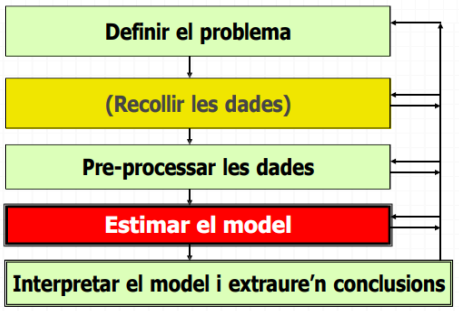
- **Sumarització:** recerca d'una descripció compacta per a un subconjunt de les dades

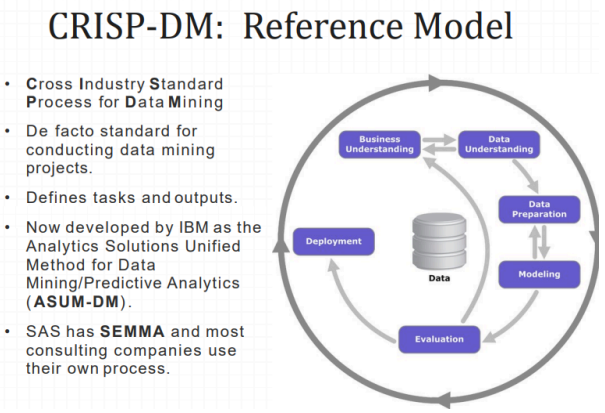
- **Detecció de canvis i desviacions:** descobrir canvis significatius en les dades

40/56

Data mining es un procés **iteratiu i interactiu**

- problema ben definit
- dades han d'estar disponibles
- genera gran quantitat dades "escombraries"
- dades ser pertinents, adequades i netes
- problema no pot ser resolt a amb consultes ordinàries o OLAP (On-Line Analytical Processing) o altres eines de bases de dades
- **Resultats han de ser validables**





<u>Raw data:</u>
Dades desconegudes, perdudes, errònies, heterogènies, diferents estructures i formats
Redundants, Irrellevants
Amb components implícits temporals i espacials (canvis de població estadística)

Les dades en el món real són "brutes“.

- **Incompletes / perdudes (missing)** : Atribut no té valors e.g. occupació=“”
- **Sorolloses (noisy):** contenen errors, alguns detectables e.g. Salari=“-10”
- **inconsistents:** Tenen discrepàncies en valors o en els codis
  - e.g. Edat=“42” Data Naixement=“03/07/1997”
  - e.g. Es valorava “1,2,3”, i ara com “A, B, C”
  - e.g. Discrepancies en registres repetits

<u>Pre Processament de dades:</u> necessari, per cada problema descrit, tenir solucions

Problemes legals de privacitat i seguretat

- que podem recollir
- que podem usar,
- la privacitat sha de mantenir en el procés
- es ètic fer servir i usar la data

Tema 2.1: Caracterització i preparació de les dades

DM molt esforç es necessita per preparar les dades per els algorismes

1. Col·lecció i selecció de dades
   Involves una base de dades i recol·lecció de techniques per obtenir les dades
2. Preparació de dades
3. Anàlisi de dades

Preparació de les dades:

Les dades poden arribar amb:

- el format erroni, irrellevant, obsoleta, noisy, equivocada, amb pèrdues, redundant

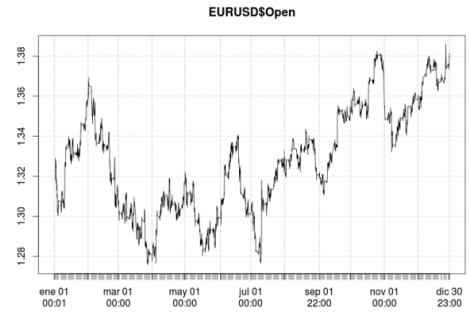
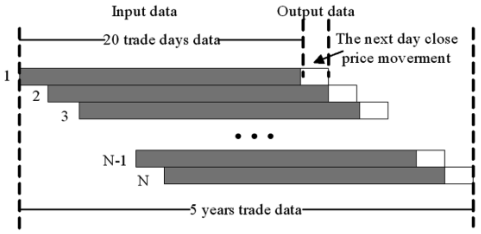
Forma Tabular:

16/76

Acme Investors Incorporated

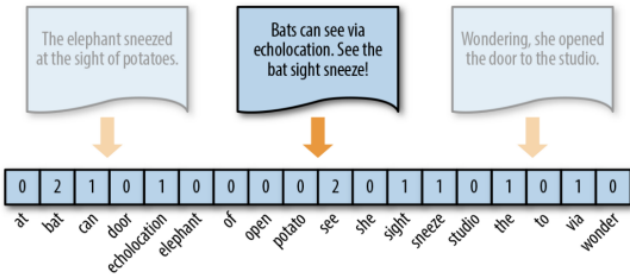
Customer ID	Account Type	Margin Account	Transaction Method	Trades/ Month	Sex	Age	Favorite Recreation	Annual Income
1005	Joint	No	Online	12.5	F	30–39	Tennis	40–59K
1013	Custodial	No	Broker	0.5	F	50–59	Skiing	80–99K
1245	Joint	No	Online	3.6	M	20–29	Golf	20–39K
2110	Individual	Yes	Broker	22.3	M	30–39	Fishing	40–59K
1001	Individual	Yes	Online	5.0	M	40–49	Golf	60–79K

Cada línia és: instant/observació/exemple
Cada columna és : característica/descripció/atribut
Standard per un algorisme DM
Les dades normalment s'extreuen en bases de dades relacionals ara en format tabular
poden ser sèries temporals,imatges transaccions,textos → transformar en taula

<table>
<tr><td colspan="2">Temporals:</td></tr>
<tr><td>evolució EUR vs USD → tabular
representació per predir el següent dia
 
</td><td>Sliding-window technique:
1) Choose size of window N (temporal context)
2) Generate rows of the table by sliding windows from beginning of temporal series to the end.
 
</td></tr>
<tr><td colspan="2">Textual Data:</td></tr>
<tr><td>Processament de documents clau DM:
1) Categorització de documents
2) Entrar altres DM com inversions
Molta info en webs,tweets,blogs..
→aconseguir l'essència del document</td><td>Bag of word technique.
1 Select a dictionary of the role features of the table
2 For each document create a row where values are the number of times each word appear in document</td></tr>
</table>

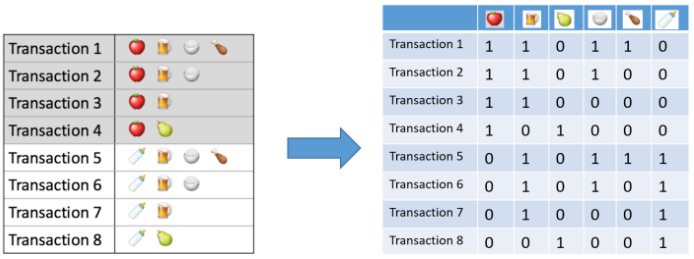
Poden haver grans numeros de columnes
i les taules molt buides

Important reduir el número de columnes
- lemmatització
- supressió de paraules *stop words*



Transactional Data:

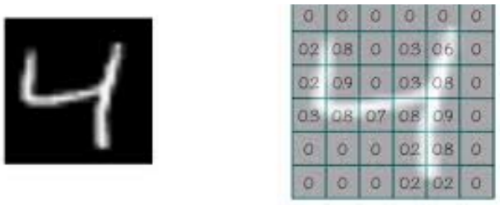
Dades trobades des de les accions o transaccions



Transaction 1	1	1	0	1	1	0
Transaction 2	1	1	0	1	0	0
Transaction 3	1	1	0	0	0	0
Transaction 4	1	0	1	0	0	0
Transaction 5	0	1	0	1	1	1
Transaction 6	0	1	0	1	0	1
Transaction 7	0	1	0	0	0	1
Transaction 8	0	0	1	0	0	1

Images:

Totes les imatges poden ser traduïdes amb números
cada píxel- expressa un grau d'intensitat



Els píxels usats com característiques per descriure una imatge per cada fila en una taula

	label	pixel0	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8	...	pixel774
0	1	0	0	0	0	0	0	0	0	0	...	0
1	0	0	0	0	0	0	0	0	0	0	...	0
2	1	0	0	0	0	0	0	0	0	0	...	0
3	4	0	0	0	0	0	0	0	0	0	...	0

<u>Entendre les dades:</u>
32/76
Categoritzar les dades entre:
1. Numeral
   1.1. Continu
   1.2. Discreta
2. Categorical: “Strings”
   2.1. Nominal: Sense cap ordre concret (colors) -inclou característiques booleanes
   2.2. Ordinals: Categories que tenen un ordre important (rangs, notes, edats..)

Numerals:
Adequades per DM algorithms: dades valuables i unitats
Estadístiques: *Average, Std, Mean, Mode, Max, Min, boxplot...*
Particular distribució que es pot verificar: *histograms, probability density estimation,*
- Analitzant parell **característiques numerals** - *scatter plots* per visualitzar dependències i correlacions per poder resumir
- Analitzant **característiques categorical** - *summary statistics* per la característica numeral per cada categoria i poder-les comparar

Categorical:

No adaptades per DM algorithms- traduïdes en numerals

Estadístiques: *Number of different categories, Frequency of each category, Most frequent category, ...*

Particular distribució que es pot verificar usant: histograms.

- Analitzant parell **característiques numerals** - *heat maps* and correlation i resumir
- Analitzant **característiques categorical** - *summary statistics* per la característica numeral per cada categoria i poder-les comparar

<u>Anàlisi de columnes:</u>

37/76

Uni-variate:

Entendre les variables que vols estudiar - trobar **estadístiques bàsiques de cada variable**

Analitzar cada columna de dades per veure les tipologies/característiques de les dades

- **quantitatives**: *max, min, median, mode, standard deviation, distribution (histograms), (boxplot), etc.*
- **qualitatives***: Number of modalities, histogram.*

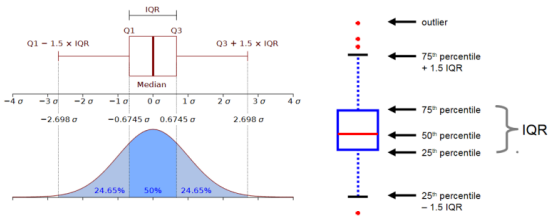
Important per detectar outliers, errors, etc

```
>>> data = pandas.read_csv('examples/brain_size.csv', sep=';', na_values=".")
>>> data
    Unnamed: 0  Gender  FSIQ  VIQ  PIQ  Weight  Height  MRI_Count
0            1  Female   133  132  124     118    64.5     816932
1            2    Male   140  150  124     NaN    72.5    1001121
2            3    Male   139  123  150     143    73.3    1038437
3            4    Male   133  129  128     172    68.8     965353
>>> data.describe(include='all')
```

	Gender	FSIQ	VIQ	PIQ	Weight	Height	MRI_Count
count	40	40.000	40.000	40.000	38.000	39.000	4.000e+01
unique	2	NaN	NaN	NaN	NaN	NaN	NaN
top	Male	NaN	NaN	NaN	NaN	NaN	NaN
freq	20	NaN	NaN	NaN	NaN	NaN	NaN
mean	NaN	117.450	112.350	111.025	151.053	68.526	9.088e+05
std	NaN	31.137	23.616	22.471	23.479	3.995	7.228e+04
min	NaN	77.000	71.000	72.000	106.000	62.000	7.906e+05
25%	NaN	90.750	90.000	88.250	135.250	66.000	8.559e+05
50%	NaN	131.000	113.000	115.000	146.500	68.000	9.054e+05
75%	NaN	138.250	129.750	128.000	172.000	70.500	9.501e+05
max	NaN	240.000	150.000	150.000	192.000	77.000	1.080e+06

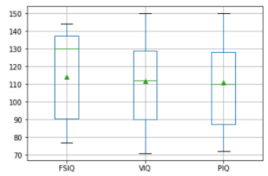
<u>Boxplot:</u>

Manera de **visualitzar info de la distribució de dades**




```
In [8]: new.boxplot(column=['FSIQ', 'VIQ', 'PIQ'],meanline=False,showmeans=True,return_type='dict')
        print("Boxplots")
        data=new
```

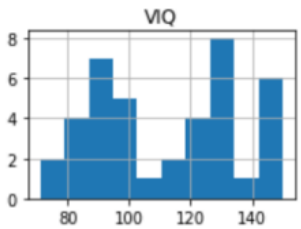
Boxplots

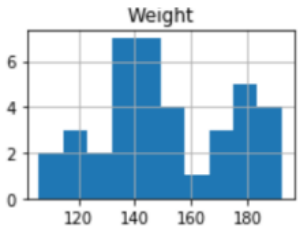


<u>Histograma:</u>

Una **visualitzar info distribució sola variable**, divideix valors per capses i ensenya per cada número d'objectes per cada capsa

- l'altura indica el num dels objectes
- Forma del histograma depen del número de capses





Estudi bi-variate:

Estudiar una variable en relació una altre

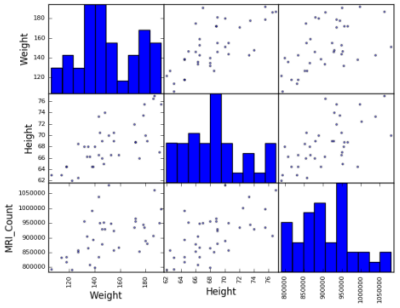
Per detectar correlacions,anti-correlacions.redundancies - per entendre les dades

<u>Dues quantitatives:</u>

Scatter plots, regression and correlation value

(tendencia a si es més pesant té més altura)

```
>>> from pandas.tools import plotting
>>> plotting.scatter_matrix(data[['Weight', 'Height', 'MRI_Count']])
```

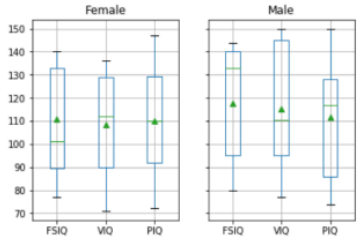


<u>Qualitativa i Quantitativa:</u>

Boxplot per cada modalitat

```
groupby_gender.boxplot(column=['FSIQ', 'VIQ', 'PIQ'],meanline=False,showmeans=True,return_type='dict')
print('Boxplots separated by Gender')
```

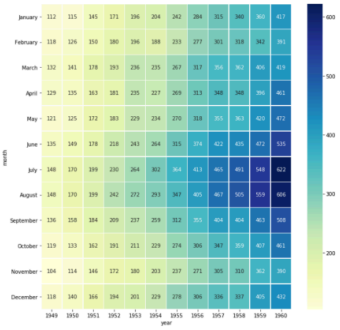
Boxplots separated by Gender



<u>Dues qualitatives:</u>

heatmap per cada parella de modalitat amb x² comparacions

```
>>> flights = sns.load_dataset("flights")
>>> flights = flights.pivot("month", "year", "passengers")
>>> ax = sns.heatmap(flights)
```



<u>Processament de dades:</u>

51/76

Dades generalment estan:

- **Incompletes**: falten valors d'atributs, falten atributs d'interès
- **Noisy**: tenen errors, inconsistencies o outliers

Tasques de processament de dades:

- **Data cleaning**: fill in missing values, remove outliers, and resolve inconsistencies.
- **Data transformation**: normalization and distribution transformation of features.
- **Data reduction**: reducing the features or rows but producing the same or even better results.
- **Data augmentation**: augmenting the number of columns using multiple databases or building new features

Data Cleaning:

<u>Missing data:</u>

Per columnes o files que no tenen valors

Culpa de: valor perdut en llegir dada, en transmissió o característica no aplicable a la dada

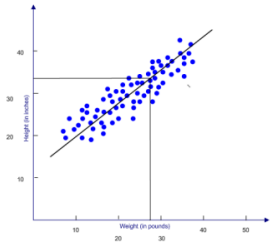
Es necessita que totes les taules estiguin plenes per treballar-hi

Solucions:

- **Borrar**: borrar fila/columna de les dades perdudes
  Pot reduir molt el valor de dades per DM
  Eliminar línies amb dades perdudes pot ser *bias* en el dataset
  Només aplicable si borrar no decrease la qualitat del dataset (no recomanat)

- **Imputation**: entrar en la dada perduda un valor que tingui sentit
  Fer sentir sentit comú- les imputations no haurien de ser bias en el dataset

  1. Canviar-ho per una **constant global:** -1 si els valors son positius, una constant que tingui sentit, una columna especial marcant el NO valor de la anterior columna
  2. Replace la **dada amb la mediana/mitjana de dades numeriques** o més comú categoricament de la qualitat per replace el valor q falta
  3. Replace la **característica amb la mitjana/median de la categoria**, si la DM es per classificar substituir per la mitjana del valor de la característica per la seva línia
  4. **Buscar correlacions** amb altres característiques i substituir els valors depenen de les correlacions (entre dues o més dades)
  5. Substituir e**l valor perdut per K casos** més similars:
     a. Aplicar una distancia en funcio de K per buscar casos
     b. Trobar la mitjana/median/mode dels K casos
     c. Substituir el valor perdut per el trobat



<u>Trobar errors:</u>

63/76
Errors consisteixen en dades que no representen la realitat
facil trobar els valors son impossibles: preus negatius, edats molt altes
Es poden detectar si estan **fora de rang** - fent una *uni-variate anàlisi* de les columnes
Si son molt improbables sabent les correlacions amb les variables- edat amb num fills
Els errors poden ser detectats pq estan **fora de la línia de regressió** de les dues variables- *bi-variate anàlisi* de columnes correlacionades
Trobat error:

1. Res
2. Eliminar la fila o columna del error (especialment si hi han molts errors)
3. Substituir el valor fent servir missing data tecniques

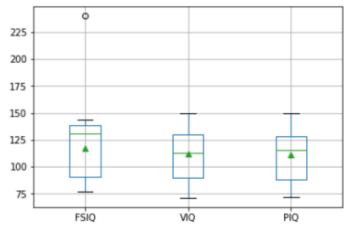
<u>Finding outliers:</u>

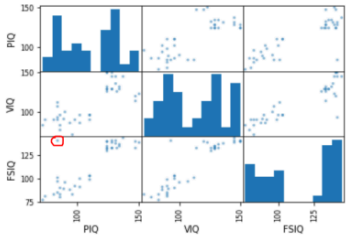
Dades que son **diferents de la normalitat** q semblen **errors**
La detecció poden ser les mateixes q detecció d'errors -majoria de casos eliminar la fila
Working assumption: considerant normals i anormals observacions en les dades

- Crear un perfil normal de característiques: *patterns o resums d'estadistiques*
- Fer servir el perfil normal per veure *outliers*: dades amb característiques molt diferents al perfil normal

1. **Gràfic:**

- Generate and visualize boxplot (1-D) and Scatter plots (2-D).





- Problems: Time consuming and subjective

2. **Statistical-based:**

Paramedic model descrivint una distribució de dades (*normal distribution*)
Depèn: distribució dades, paràmetres de distribució*(mean/variance)* i num expected outliers

- Example: Univariate Outlier detection

  $Age = \{3, 56, 23, 39, 156, 52, 41, 22, 9, 28, 139, 31, 55, 20, -67, 37, 11, 55, 45, 37\}$

- Statistical parameters are:

$$\mu = 39.9$$

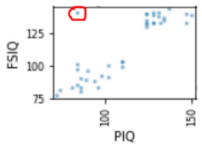
$$\sigma = 45.65$$

- If we select that the threshold value for normal distribution of data is

$$Threshold = \mu \pm 2\sigma$$

- then all data out of range [-54.1, 131.2] will be potential outliers: $\{156, 139, -67\}$

- Example: Bivariate Outlier detection



- When correlation between variables and one point is far from the regression line, then it could be an outlier.
- Procedure:
  1. Compute regression line
  2. Compute standard deviation of errors (distances of points to regression line)
  3. Find points that are further than k times the standard deviation of the regression line

3. **Distance-based i Density-based:**

Trobats per anàlisi uni-variate o bi-variate però alguns poden escapar el anàlisi
Es poden trobar **analitzant línies** en comptes columnes
Principal esq un **outlier estarà lluny dels altres exemples**/ estara en **zona de poca densitat**

- **Distance approach.** *An instance o in a data set S is an outlier if its k-th nearest neighbor is at a too high distance when compared with rest of examples of S.*

- **Density approach**: *An instance o in a data set S is an outlier if at least a fraction p of the samples in S lies at a distance greater than d*

76/76

Tema 2.2: Transformació de Dades

Value transformation:
Canviar valors de la taula

Homogeneïtzació de les dades:

Categorical a Numeral:

alguns DM algorithms només funcionen sent numerals:

- **Ordinal (Label) encoding:**
  Categories que tenen un ordre implicit - passar el rang literal en rang numeral

Coding in python and pandas:

```
Age_dict = {'Child':1, 'Young':2, 'Adult':3, 'Old':4}
df['Age'] = df.Age.map(Age_dict)
```

Example of transformation for *Age* attribute:

	...	*Age*	...
o_1	...	Child	...
o_2	...	Young	...
o_3	...	Adult	...
o_4	...	Child	...
o_5	...	Old	...
...	...	...	...

$\Longrightarrow$

	...	*Age*	...
o_1	...	1	...
o_2	...	2	...
o_3	...	3	...
o_4	...	1	...
o_5	...	4	...
...	...	...	...

- **One-Hot encoding:**
  Categories no tenen ordre relacionat al target i no hi han moltes categories(ex.color)
  NO correcte introdueix diferents instàncies entre les modalitats
  Crear ***dummy variable*** per cada modalitat (categories sempre en **binari** 0/1)
  → si masses modalitats pot agrupar valors per reduir el número- normalment si hi ha algun tipus de **jerarquia** estructura sobre(tipus de malaltia)
  dummys:

	...	*color*	...
o_1	...	red	...
o_2	...	blue	...
o_3	...	brown	...
o_4	...	red	...
o_5	...	green	...
...	...	...	...

$\not\Longrightarrow$

	...	*color*	...
o_1	...	1	...
o_2	...	2	...
o_3	...	3	...
o_4	...	1	...
o_5	...	4	...
...	...	...	...

	...	*color*	...
o_1	...	red	...
o_2	...	blue	...
o_3	...	brown	...
o_4	...	red	...
o_5	...	green	...
...	...	...	...

$\Longrightarrow$

	...	*red?*	*blue?*	*brown?*	*green?*	...
o_1	...	1	0	0	0	...
o_2	...	0	1	0	0	...
o_3	...	0	0	1	0	...
o_4	...	1	0	0	0	...
o_5	...	0	0	0	1	...
...	...	...	...	...	...	...

- **Mean encoding:**
  Objectiu **preedir la etiqueta**: modalitats poden ser transformades segons ajuntar diferents etiquetes

	...	*color*	...	*label*
o_1	...	red	...	yes
o_2	...	blue	...	yes
o_3	...	brown	...	no
o_4	...	red	...	no
o_5	...	green	...	yes
...	...	...	...	...

$\Longrightarrow$

	...	*color*	...	*label*
o_1	...	0.5	...	yes
o_2	...	1	...	yes
o_3	...	0	...	no
o_4	...	0.5	...	no
o_5	...	1	...	yes
...	...	...	...	...

Numerical to Qualitative:

Reduir el nombre de rang (ex: age: 0-18, 19-40, 40-65, 67-120)
Passar a un qualitatiu (age: young, adult, senior, old)

Reescalar les dades:

17/78

Totes variables haurien estar dins del **mateix rang** per evitar computation of distances

Mandatory for some algorithms like *k-NN, SVM or Neural Networks*

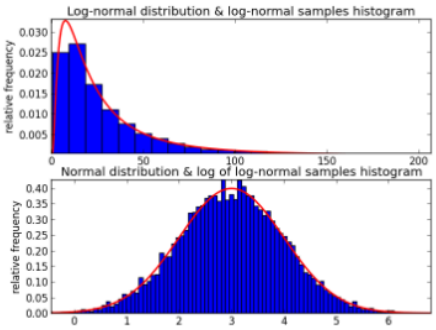
• Normalització [0-1]	• Standardization N(0,1)

$$v(i) = \frac{v(i) - min(v)}{max(v) - min(v)}$$ `from sklearn.preprocessing import MinMaxScaler`
`scaler = MinMaxScaler()`
`scaler.fit(data)`
`d2 = scaler.transform(data)`
	$$v(i) = \frac{v(i) - mean(v)}{std(v)}$$ `from sklearn.preprocessing import StandardScaler`
	`scaler = StandardScaler()`
	`scaler.fit(data)`
	`d2 = scaler.transform(data)`

Canviar la distribució:

En alguns casos de distribució de dades tenen *heavy tails*-> problemes distingint entre moltes dades -> les **dades** apareixen **com *outliers***

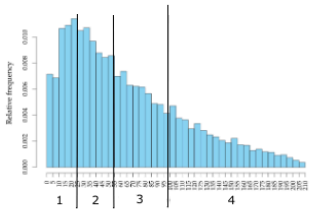
Effect of log-normalization:



<u>Reemplaçar per quantiles:</u>

Reemplaçar el valor segons el quantile

Efecte de separació de quantiles fent servir 4 valors (es pot usar més si es necessita)



<u>Treball en les columnes:</u>

24/78

Reduir el número de columnes: Treure variables

Molt comú DM⇒ redueix computing time, millora predictiu/descriptius resultats, millora la representació del model DM, la dimensió intrínseca pot ser petita

Algunes tècniques DM poden no ser efectives en high-dimensional data

- **<u>Feature Selection:</u>**

Procés q tria un **subset òptim** de característiques acord una **funció objectiva**

Algorismes de filtrar i rankear les categories

- ➔ Remove columns without enough variability (std(v) < threshold)
- ➔ Remove variables that are not enough correlated with label (corr (v , label) < threshold)
- ➔ Remove variables with low Fisher score with label (F (v , label) < threshold)
- ➔ Remove variables without enough Information Gain (IG (v) < threshold)

➔ Remove variables without enough Relief [Paper] (Relief (v) < threshold)

Tots aquests mètodes implícitament creen **ranking de categories** donades unes mesures, després del ranking s'usa un ***threshold*** per seleccionar les **característiques interessants**

1) Ranking o filter methods:

Tenint x_V^L els valors de categories V amb la etiqueta l i $n_{V,l}$ el número de valors del set

- *Fisher Score* troba diferències de valors per diferents característiques (col.)

$$F_v = \frac{|\,(\mu(X_v^1) - \mu(X_v^0)\,|}{\sqrt{\frac{\sigma(X_v^1)}{n_{v,1}} + \frac{\sigma(X_v^0)}{n_{v,0}}}}$$

- Example:

v_1	v_2	l
0.3	0.7	0
0.2	0.9	1
0.6	0.6	0
0.5	0.5	0
0.7	0.7	1
0.4	0.9	1

$$X_{v_1}^0 = \{0.3, 0.6, 0.5\}$$
$$X_{v_1}^1 = \{0.2, 0.7, 0.4\}$$
$$X_{v_2}^0 = \{0.7, 0.6, 0.5\}$$
$$X_{v_2}^1 = \{0.9, 0.7, 0.9\}$$

$$\sqrt{\frac{\sigma(X_{v_1}^1)}{n_{v_1,1}} + \frac{\sigma(X_{v_1}^0)}{n_{v_1,0}}} = \sqrt{\frac{0.0233}{3} + \frac{0.6333}{3}} = 0.4678$$

$$\sqrt{\frac{\sigma(X_{v_2}^1)}{n_{v_2,1}} + \frac{\sigma(X_{v_2}^0)}{n_{v_2,0}}} = \sqrt{\frac{0.01}{3} + \frac{0.0133}{3}} = 0.0875$$

- Test with threshold 0.5:

$$\frac{|\,(\mu(X_{v_1}^1) - \mu(X_{v_1}^0)\,|}{\sqrt{\frac{\sigma(X_{v_1}^1)}{n_{v_1,1}} + \frac{\sigma(X_{v_1}^0)}{n_{v_1,0}}}} = \frac{|\,0.4333 - 0.4667\,|}{0.4678} = 0.0735 < 0.5$$

$$\frac{|\,(\mu(X_{v_2}^1) - \mu(X_{v_2}^0)\,|}{\sqrt{\frac{\sigma(X_{v_2}^1)}{n_{v_2,1}} + \frac{\sigma(X_{v_2}^0)}{n_{v_2,0}}}} = \frac{|\,0.8333 - 0.6\,|}{0.0875} = 2.6667 > 0.5$$

- Correlation:

$$\rho(v, l) = \frac{cov(v, l)}{\sigma_v \sigma_l}$$

- Information Gain (Mutual Information): when v and l are independent variables $p(v, l) = p(v)p(l)$

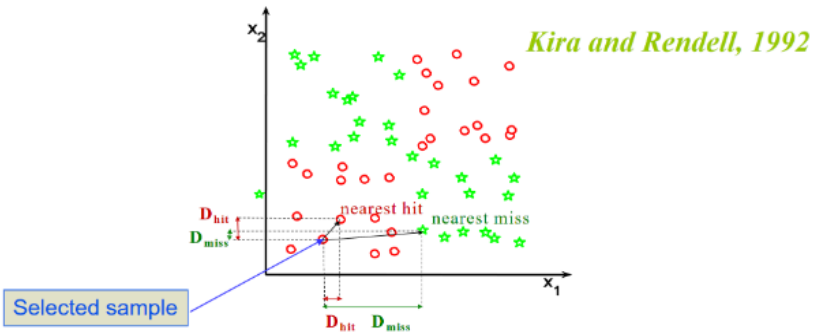
$$I(v; l) = \sum_{x \in D} p(x_v, x_l) \log \frac{p(x_v, x_l)}{p(x_v)p(x_l)} = H(l) - H(l|v)$$

where H is entropy measure

- *Relief algorithm:*
  - Construcció **Algoritme basic:**
    - Each feature is assigned with *cumulative weight* computed over a subset of the *training data set.*
    - Feature with weight over a *certain threshold* is the selected feature subset.
  - Weight computation:
    - Let X be a *randomly* selected sample
    - *near-hit sample* = closest sample from the same class
    - *near-miss sample* = closest sample from the different class.
    - Impact on weight for feature j of example X is computed as:

$$W_j = -diff_j(X, near\text{-}hit)^2 + diff_j(X, near\text{-}miss)^2$$

    - Procedure repeated for several random samples



2) Redundant features:

Eliminar característiques redundants (similars valors entre elles **després** de la **normalització**)- si redundant eliminar una

Però el temps consumit per testear cada parella de caracteristques cost O(f²)

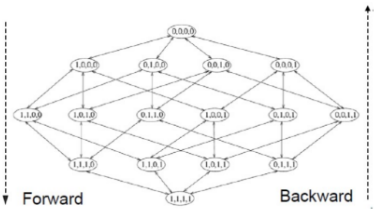
Normalment no s'interposen en els models,nome en l'eficiència i interpretació

- **<u>Feature Extraction:</u>**

39/78

Com triar un threshold?

- **Rankear** característiques i usar **top n** en el rànquing
- **Plot característiques** en contrari els valors i triar threshold **quan valors decreixent**

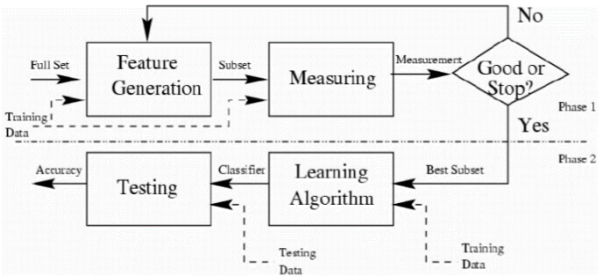
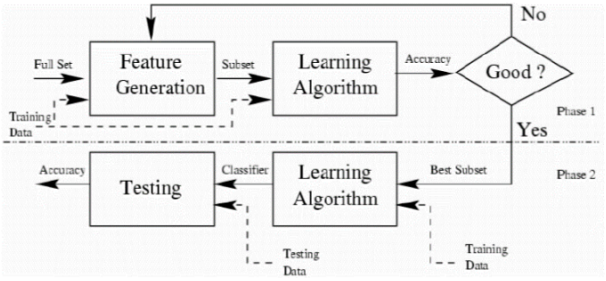


Evaluar sets de característiques (no característiques indep) ------> problema de busqueda (2^n sets de característiques a considerar)

Busqueda optima no possible, **simplificacions** per aconseguir un temps raonable:

Feature set generator que proposa set de característiques per evaluar

- Mètode d'avaluació:
  - **FILTER**: **metrica d'avaluació independent** del mecanisme d'aprenentatge que es farà servir
  - **WRAPPER METHOD:** usa **mesures de l'objectiu d'aprenentatge** per avaluar el set de característiques

Filter method	Wrapper method:
	

i un criteri per parar la búsqueda en el set de característiques

Reduir número de columnes:

1) <u>Feature selection</u>: tria subset òptim segons les característiques en **funció objectiva**
   - *Feature ranking algorithms*
   - *Minimum subset algorithms*
2) <u>Feature extraction</u>: **mapping la high dimensional data** en un **lower dimensional data**

Les noves dimensions(columnes) son **NOVES** i tenen info del dataset original en un número menor de columnes

a) **Descriptive setting: PCA**: minimitzar la pèrdua d'informació sent les columnes **combinacions lineals** de les originals, ordenades per inertia, només guardar n columnes amb més informació
b) **Predictive setting: LDA**: mateix q descriptive setting, però **maximitza la discriminació entre classes,** (comptes de minimitzar pèrdua dades)

Enriquir el dataset: Ajuntar taules

49/78

Normal tenir info en dataset, en comparació mètodes relacionals, la info es guarda en diferents taules- pots **completar(enriquir)** el **data set amb una altre taula**

Ex: Taula de malalties i una altre taula de id de pacients -> ajuntar 2 taules es poden relacionar característiques entre la malaltia i característiques dels pacients

Enriquir el dataset: Crear noves variables

Si dades expressades en columnes difícils de manipular en DM - definir **noves columnes**

Per enriquir el dataset per fa facilitar aconseguir un millor model

- Fent **ràtios i proporcions** de dades del dataset
- Creació automatica:

SI algorismes només funcionen en dependències lineals de variables -> generar nova característica amb totes les **combinacions polinominals** amb un grau menys o igual a un grau determinat

$$\{X_1, X_2, X_3\}$$

add set of new variables

$$\{X_1X_2, X_1X_3, X_2X_3, X_1X_2X_3\}$$

with a degree of 3.

<u>Treball en les files:</u>

56/78

Reduir el nombre de files:

Molts DM algorismes temps cost: O(n²)(matrius) O(n³) si n= num files del dataset

En alguns datasets es pot tenir tantes files q no es pot executar en un temps raonable

<u>Sampling:</u>

- Quants exemples entren en el sample?/ quin tipus/ pot ser sample correcte per db?

1) **Systematic sampling**: cada k-th sample en el dataset, si vols 50% del sample tria un i el següent no
   -><u>Problema</u>: si hi han regularitats en el dataset (ex. ordenats)
2) **Random sampling:** random sampling sense reemplaçar o amb remplaços
   -><u>Problema</u>: El sample és representatiu del dataset complet?
3) **Stratified sampling:** dividir el dataset en no-duplicats subsets *strata* i combinar els resultats dels strata
   -><u>Problema</u>: ok. valors de classes son representades. q passa altres columnes?
4) **Multi-Stratified sampling**: fer **stratified sampling**, després verificar que cada columna tenen estructures **estadístiques similars del dataset original,** si el sample no passa el test->borrar i generar un altre sample fins q no passi el test

Nombre samples:

Error seria fer servir el dataset complet- representatiu al dataset

- **incrementar el sampling**: començar petit i **medir la correctesa** (accuracy) amb ia i anar **augmentant** el tamany del sample fins q **no es troben més millores**
- **Inverse sampling:** fent servir característiques rares *(skewed distributions)* consisteix en incrementar el tamany del sample fins que alguna **condició d'alguna característica sigui certa** (ex: tots els valors apareixen al menys t vegades)

Assume a SVM with training cost $O(n^3)$ and $n = 10^6$ examples:

$$O((10^6)^3) = 10^{18}$$

Consider the following procedure:

- Divide the dataset in 10 parts
- Build an SVMs on each part (we have 10 SVMs)
- Combine by *majority vote* the output of each classifier in testing
- Now training cost is:

$$\sum_{i=1}^{10} O((10^5)^3) = 10^{16}$$

Per no tirar els exemples del dataset -> ajudar pel cost polynomial -> *ensemble techniques*

Incrementar el nombre de files:
69/78
Si en el cas de no tenir moltes dades (petits o no balancejats datasets) - generar data

<u>Datasets no balancejats:</u>
Difícil obtenir bons resultats pq hi han classes no representades -DM *biased* per una classe

- La *accuracy* és mentida en aquests casos-> resultats de *precision/recall/F-measure*

1) **Undersampling**: representació similar de la classe petita eliminant exemples de les classes més populars
2) **Oversampling**: representació similar de la classe petita repetint el exemple del dataset varies vegades de les classes menys populars
3) **Generar nous exemples de la classe petita:** ex SMOTE , tria aleatoriament 2example de les classes petites i troba mitjana -> incloure en la classe petita
   Repetir fins que hi havien prous examples per la classe petita

Tema 3: Mètodes supervisats: Intro i avaluació

1/85

Donat un dataset D d'observacions, normalment es pot representar com una taula per cada fila es una observació/ columna una descripció

→ Arribar a predir una columna del dataset anomenada *label* per dades noves

(ex: dataset emails si spam o no, diagnostics si es escisofrenic o no)

Per classificar un objecte en una classe correcte - construir un **classificador (classifier)**

(Considerar q dataset contenen positius/negatius exemples per cada label)

Generalitzar habilitat d'extendre coneixement del dataset en dades noves → **SIMPLICITAT**

Si *learning method* massa flexible pot tenir risc **overfitting les dades**

-> learning mth. tenen tendencia a simplificar les explicacions de les dades:

- Shortest description:
  - Minimum description length
  - Shortest tree size
  - Fewest parameters model
  - Independence of columns
- Regularization about complexity of the classifier:
  - Values of parameters (f.i. limited weight in neural networks)
  - Trade-off on capacity of classifier and error

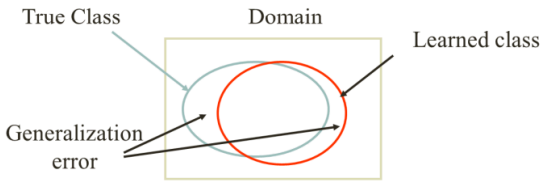
<u>Avaluació de classificadors:</u>

24/85

Introducció i mesures:

DM té un mode supervisat té una columna **TARGET**

diff de clustering - tenim una **forma objectiva** de **mesurar la *success* dels classificadors**- provar en testos els nous casos fets pel classificadors i verificar columnes creades/realitat



Generelització d'error - part de la **classe real** i part de la **classe nova creada- learned class**

$$E_{empiric} = \frac{1}{n}\sum_{i}^{n}\left\|\{i \mid H(x_i) \neq C(x_i)\}\right\|$$

Donat un classificador, test els errors que produeix i crear una **CONFUSION MATRIX:**

	Actually Positive	Actually Negative
Predicted Positive	True Positive cases (TP)	False Positive cases (FP)
Predicted Negative	False Negative cases(FN)	True Negative cases (TN)

Cada entrada és el nombre d'instàncies

ACCURACY: Error empric del set de cados donat: $Ac = \frac{TP + TN}{TP + TN + FP + FN}$

No una bona mesura quan:

Datasets no balancejats :

	Act. Pos.	Act. Neg.
Pred. Pos.	10	100
Pred. Neg.	40	300

	Act. Pos.	Act. Neg.
Pred. Pos.	50	140
Pred. Neg.	0	260

Unbalanced case: Assume we are interested in detecting one kind of documents from a large corpus. Both tables same accuracy but show different behaviors.

Diferents costs:

	Act. Pos.	Act. Neg.
Pred. Pos.	100	10
Pred. Neg.	100	290

	Act. Pos.	Act. Neg
Pred. Pos.	100	100
Pred. Neg.	10	290

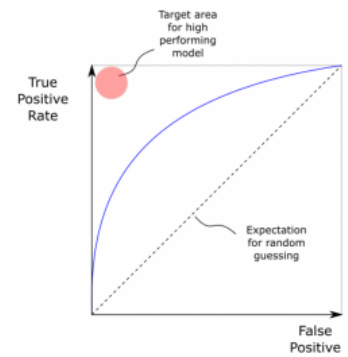
Different costs: Assume cost of misclassifying a positive case is 10 times misclassifying a negative case. Same accuracy but different cost.

RECALL: Proporció de casos positius que son detectats: $R = \frac{TP}{TP + FN}$

PRECISIÓ : Proporció casos positius respecte num casos numerats positius: $P = \frac{TP}{TP + FP}$

F-MEASURE : Harmonic mean of **RECALL** and **PRECISION**: $P = \frac{2}{1/P + 1/R} = \frac{2\,PR}{P + R}$

ROC curve: si un classificador retorna una *belief en la predicció*, obtenir diferents prediccions canviant el cut-off point, Plot de **ROC curve (FP vs TP rate)**



- Are Under Curve (AUC) measure to compare classifiers
- Interesting to find best threshold for classifier
- ...specially good for imbalanced data sets

Cross-validation:

37/85

Matrix confusion es generada fent test del classificador de les noves dades

No es pot fer servir els **exemples usats** per **crear el classificador** per generar la matriu (**biased**)

→**NO** fer servir totes les dades per el learning, **guardar** alguns exemples per **testing**

Dividir la dades en 2 sets learning/testing

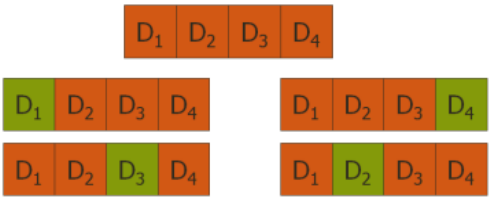
66% dades per learning i 33% data testing →

<u>Inconvenients</u>:

- Si dades son poques, no dividir per testing
- Els resultats poden dependre de divisió de les dades →dividir vàries vegades fent mitjana de resultats

K-fold Cross-validation:

47/85



Dividir el dataset en K subsets de mateixa longitud:
Repetir **k** vegades el **learning process**, cada vegada **deixant** un **subset** per testing
Average the K accuracy estimators
K es intercanvi temps/variança : entre 5 o 10

<u>Avantatges</u>:
- el estimadors més robustos (menys variança)

<u>Inconvenients</u>:
- Es perden 1/k dades per testing
- Repetir el learning process k times

Leave-one-out (LOO):

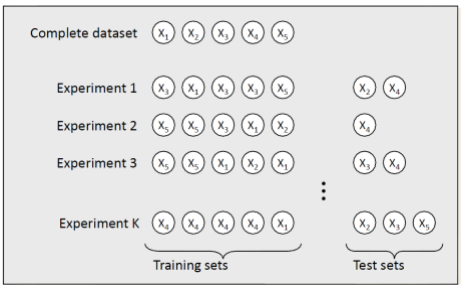
Versió extrema **<u>k-flold cross-validation,</u>** on **K es N**, el número d'exemples al dataset
Repetir **N execucions del learning algorithm,** amb **N-1** exemples per **learning** i **1** per **testing** (cada vegada 1 diferent)
Average the n estimators

<u>Avantatges</u>:
- Màxim profit als exemples
- Redueix variança del estimador

<u>Inconvenients</u>:
- Més temps de compute

Bootstrapping:

51/85



Repetir el procés K vegades(centenars):
- **Aleatòriament** selecionant **n examples** pel **training**
- **Exemples no seleccionats** pel training usats per **testing**

El true error es la mitjana del error de test data.

Confidence intervals:

$E_{empirical} \rightarrow \acute{\varepsilon}$ segueix una **distribució binomial** amb mean $\acute{\varepsilon}$ **i** $\sigma = \sqrt{\frac{\acute{\varepsilon}\,(1-\acute{\varepsilon})}{n}}$

Quan n $\acute{\varepsilon}$ (1-$\acute{\varepsilon}$) >= 5 binomial es pot aproximar a **distribució Normal** amb mateixa mean i variança: $\acute{\varepsilon} - z_{\alpha/2}\,\sigma <= \varepsilon <= \acute{\varepsilon} + z_{\alpha/2}\,\sigma$ 95% confidence $z0.025 = 1.96$

Quan n $\acute{\varepsilon}$ (1-$\acute{\varepsilon}$) < 5 aplicar **cumulative distribució de binomial**, probabilitat de fer màxim k errors: $Pr\{X <= k\} = \sum_{i=0}^{k} (n\,i)\ \acute{\varepsilon}^{i}\,(1-\acute{\varepsilon})^{n-1}$

Els binomials es poden aproximar a **Normal** $\acute{\varepsilon} - z_{\alpha/2}\,\sigma <= \varepsilon <= \acute{\varepsilon} + z_{\alpha/2}\,\sigma$

Però ara $\acute{\varepsilon}$ es mitjana de totes les cross-validation i $\sigma = \sqrt{\frac{\acute{\varepsilon}\,(1-\acute{\varepsilon})}{n}}$ on n es el nombre de cross-validation runs

Resultat: decidir quins dels dos classificadors h0 i h1 te el **menor error rate**

Creen la següent taula:

n_{00}: number of cases both right	n_{01}: number of cases h_0 right and h_1 fails
n_{10}: number of cases h_0 fails and h_1 right	n_{11}: number of cases both fail

Comparar els dos classificadors McNemar's Test

Null hypothesis: $n_{01} = n_{10}$ (both same error)

McNemar's Test:

$$M = \frac{(|n_{01} - n_{10}| - 1)^2}{n_{01} + n_{10}} > \chi^2_{1,\alpha}$$

M is distributed approximately as χ^2 with 1 degree of freedom. For a 95% confidence test, $\chi^2_{1,0.95} = 3.84$. So if M is larger than 3.84, then with 95% confidence, we can reject the null hypothesis

Once we know they don't have the same error, which one is better?

<u>Multi-class classification:</u>

63/85

En algunes característiques de classificadors assumim es binari : +1 /-1

Exemples positius etiqueta +1 / ex negatius etiqueta -1

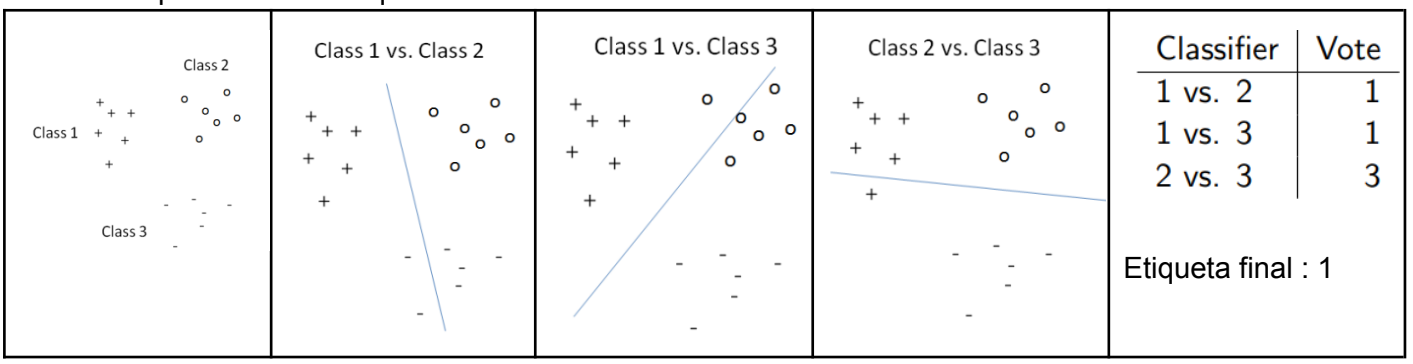
Si les etiquetes **no son binaries** podem crear una **combinació de classificadors**:

One vs One(OVO)

Given k labels, learn to separate class 1 from 2, class 1 from 3, ... class 1 from k, class 2 from 3, class 2 from 4, ... class 2 from k, ... class k − 1 from class k

Després *learing k(k−1) / 2* classificadors, **aplicar tots** els **classificadors** i **votar**

Si empat- **triar** una etiqueta aleatòriament



Classifier	Vote
1 vs. 2	1
1 vs. 3	1
2 vs. 3	3

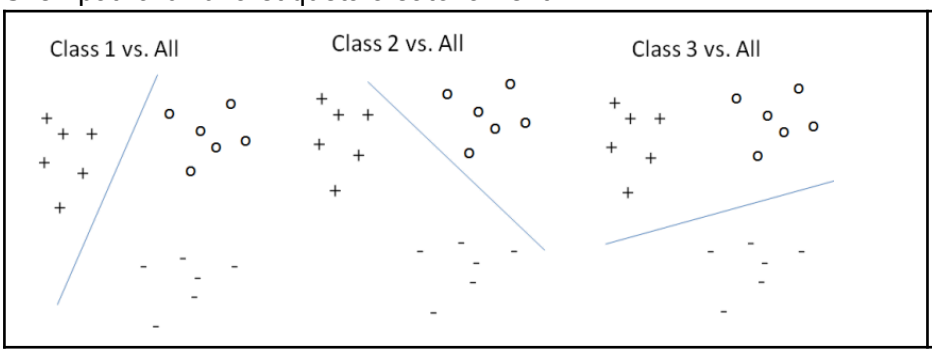
Etiqueta final : 1

One vs All (OVA)

Given k labels, learn to separate class 1 from all other classes, class 2 from all other classes .... class k − 1 from all other classes.

Després *learing* k classificadors, quan **etiquetar nova observació**, aplicar **tots els classificadors i votar**

Si empat- **triar** una etiqueta aleatòriament



Classifier	Vote
1 vs. All	1
2 vs. All	All
3 vs. All	All

Etiqueta final : 1

Data preparation:

70/85

A partir d'ara assumim temim **dataset NET**

- **No falten valors**
- **Característiques rellevants** incluides al dataset (amb no moltes més columnes)
    - Algorismes funcionen **num moderat columnes irrellevants**- si hi han masses algorisme pot fallar
    - Alguns mètodes eliminen característiques irrellevants (**feature selection**)
- Alguns algorismes només funcionen amb **numerical o categorical data**
    - Crear noves columnes per cada categoria

Object	Color
Obj1	blue
Obj2	green
Obj3	green
Obj4	red

$\Rightarrow$

Object	blue	green	red
Obj1	1	0	0
Obj2	0	1	0
Obj3	0	1	0
Obj4	0	0	1

    - Discretització de dades continues:
        - *Rounding - decimation*
        - *Best cut places -Minimize*

$$ER = \sum_{n=1}^{k} \left(\sum_{i \in bin_k} |v_{i,n} - v_{\mathrm{moda},n}| \right)$$

where k is the number of bins (categories). For instance {1, 1, 2, 2, 2, 5, 6, 8, 8, 9} into three bins: **A**={1, 1, 2, 2, 2}, **B**={5, 6}, **C**={8, 8, 9}

        - *Target oriented methods* : stadistiacment millor manera de tallar les dades depenen de l'etiqueta *ChiMerge method*
- intentar treballar amb **dades normalitzades o estandarditzades**
    - Donar el mateix pes a cada característica per fer-les unitats independents

Normalization 0,1:

$$v_i^n(k) = \frac{v_i(k) - min(v_i)}{max(v_i) - min(v_i)}$$

Standardization to mean 0, sigma 1

$$v_i^n(k) = \frac{v_i(k) - \mu(v_i)}{\sigma(v_i)}$$

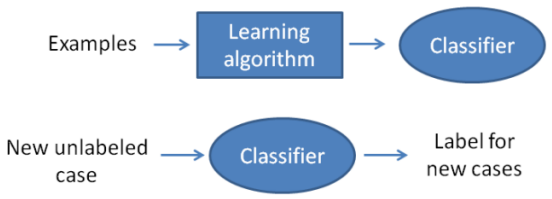
Generalitzacions al crear classificadors:

82/85

Com determinar la millor etiqueta donades dades → **Learning algorithms**

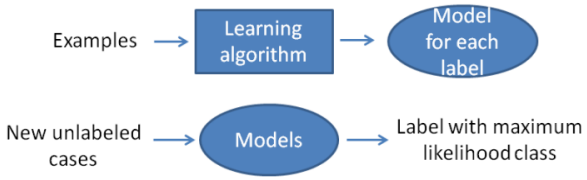
Discriminative method:

Mecanisme dona passa a **discriminar/separar casos** de diferents classes sense tenir **cap info de les dades**



Generative method:

Assumir **distribució paramètrica** de les dades- aprendre la distribució dels paràmetres i trobar el ***most likely class*** per les noves observacions



85/85

Tema 4.1: knn i Naive Bayes

1/35

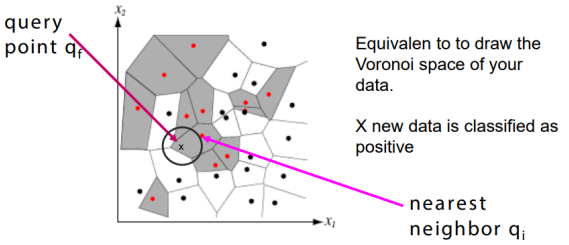
Nearest Neighbor:

Non parametric method: vaga instancia base de *learning method* no construeix cap model
Assigna etiqueta a una observació depenen de etiquetes més aprop
Necessita:

- training set
- Similarity measure

K-NN:

K- nearest neighbor algorithm
Interpreta cada exemple com **punt** en **espai** definit per **característiques** descrivint les dades. Una **similarity measure** ajuda a classificar els nous exemples
Class assignada depenent de K més aprop exemples



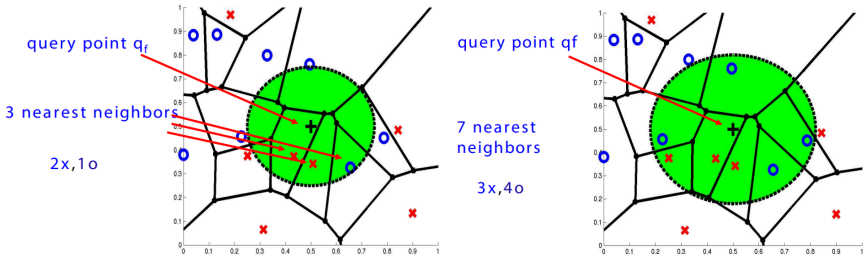
Dos característiques (x1,x2) defineixen espai
Cada punt vermell positiu// punt negre negatius

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{m} (x_i(k) - x_j(k))^2}$$

Mesures de distància:

Distància paràmetre d'un algoritme- normalment **euclidià**
CAVEAT: dades normalitzades/estandarditzades per assegurar-se de la rellevància igual per cada característica per trobar la distància correcte

Avantatges: ràpid educar (training), Habilitat aprendre funcions complexes
Inconvenients: lent en testejar
necessita una estructura complexa per representar dades en arbres
Fooled per noise i característiques irrellevants



Algoritme:

Paràmetres:

- Natural number k (odd number)
- Training set
- Distance measure

Algoritme:

1. Store all training set <xi, label(xi)>
2. Given new observation, xq, compute the **nearest k neighbors**
3. Let **vote** the nearest k neighbors to **assign the label to the new data.**

Seleccionar k:

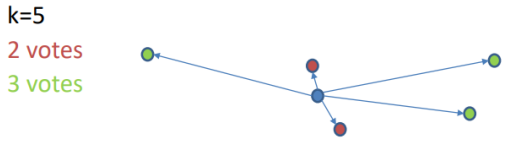
- Numeros alts: *Smother frontiers, Reduces sensibility to noise*

→Massa alts: *loose locality very distant points can interfere in assigning labels, more Computation time*

Triat per **cross-validation**

Distància amb pes:

Variació Knn- Quan voten tots els k neighbors tenen la mateixa influencia → però si estan més lluny tindrien menys influencia → donar **més pes** als **exemples més propers**



pes:

$$w_i = K(d(x_i, x_q))$$

xq: query point
d: distància de la funció
xi: i-més a prop exemples
K: kernel- decreix respecte distancia

Predicció etiqueta xq:

$$\text{sign}\left(\sum_{i=1}^{k} w_i l(x_i)\right)$$

l(xi) és {-1,1} de l'etiqueta xi,
wi pes de l'exemple xi

<u>exemple</u>:

$$\text{sign}\left(0.9 * 1 + 0.8 * *1 + 0.4 * (-1) + 0.35 * (-1) + 0.3 * (-1)\right) = \text{sign}(0.65) = +1$$

Intentar solucionar problema de *fooled by irrelevant features*→ **feature selection**
fent servir la distància amb pes:

$$d_z(x_i, x_j) = \sum_{k=1}^{n} z_l (x_i(k) - x_j(k))^2$$

limita pes 0.1 si **zj=0 ELIMINA característica**
Si troba pes zi.. zn (per una característica) minimiza el error fent **cross-validation**

20/35

Naïve Bayes:
20/35
Parametric: construeix model probabilitat de les dades usant assumptions

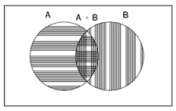
Podem estimar la probabilitat de les dades $p(x_1, x_2, \ldots x_n | c_i)$
probability to observe example with features (x1,x2... xn) [xi , represents feature i of observation x] in class ci
Per classificar una observació (x1,x2..xn) buscar la classe que maximitza la probabilitat de l'observació que pertany a la classe:

$$c_{MAP} = \underset{c_j \in C}{\operatorname{argmax}} P(c_j \mid x_1, x_2, \ldots, x_n)$$

teorema de Bayes:

$$c_{MAP} = \underset{c_j \in C}{\operatorname{argmax}} P(c_j \mid x_1, x_2, \ldots, x_n) = \underset{c_j \in C}{\operatorname{argmax}} \frac{P(x_1, x_2, \ldots, x_n \mid c_j) P(c_j)}{P(x_1, x_2, \ldots, x_n)}$$

Bayes' theorem :



$$\begin{cases} P(A|B) = P(A \wedge B) / P(B) \\ P(B|A) = P(A \wedge B) / P(A) \end{cases}$$

$$\Rightarrow P(A \wedge B) = P(A|B)\, P(B) = P(B|A)\, P(A)$$

$$\Rightarrow \boxed{P(A|B) = \frac{P(B|A)\, P(A)}{P(B)}}$$

P(cj)- proporció simple elements classe j
P(x1,x2.. xn|cj) -> només per estimar datasets gegants PROBLEMA →
<u>independence assumption :</u> attribute values son independents

$$P(x_1, x_2, \ldots, x_n \mid c_j) = \prod_i P(x_i \mid c_j)$$

P(xk|cj) : necessitem n.|C| estimacions de probabilitat → num de valor amb la propietat xk en la classe cj en diferencia al nombre complet de la classe cj

$$\boxed{c_{NB} = \underset{c_j \in C}{\operatorname{argmax}} \frac{P(x_1, x_2, \ldots, x_n \mid c_j) P(c_j)}{P(x_1, x_2, \ldots, x_n)} = \underbrace{\underset{c_j \in C}{\arg\max}\, P(c_j) \prod_i P(x_i \mid c_j)}_{\text{Equation to be used}}}$$

Probabilitats rang 0..1 problemes de *floating-point underflow* errors

$$\boxed{c_{NB} = \underset{c_j \in C}{\operatorname{argmax}} \left(\log P(c_j) + \sum_{i \in positions} \log P(x_i \mid c_j) \right)}$$

Retronen una accuracy prou bona *independence assumption* is not fulfilled
Learning and testing time son linears amb nombres i attributs

Classificar textos:

- Training set: X document corpus
- Each document is labeled with f(x)=*like/dislike*
- Goal: Learn function that permits given new document if you like it or not.
- Questions:
  - How do we represent documents?
  - How to compute probabilities?
- **How do we represent documents?**
  - Each document is represented as a *Bag of Words*
  - Attributes: All words that appear in the document
  - So each document is represented as a boolean vector with length N: 0 – word does not appear ; 1 – word appear
- Practical problem: A very huge table.
- Solution : Use sparse representation of matrixes

- Some numbers
  - 10.000 documents
  - 500 words per document
  - Maximum theoretical number of words: 50.000 (much less because of word repetitions)
- Reducing the number of attributes
  - Removing the number (sing/plural) and verbal forms (*stemming*)
  - Remove conjunctions, propositions and articles (*stop words*)
  - Now we have about. 10.000 attributes

$$\mathbf{v_{NB} = \underset{v \in \{like, dislike\}}{argmax}\ P(v) \prod_i P(x_i = word_i \mid v)}$$

How to compute probabilities?

- First compute $P(v_i)$ for each class ["a priori" probability for like and dislike classes]

$$P(v_{like}) = \frac{\#\text{documents like}}{\text{total number of documents}}$$

$$P(v_{dislike}) = \frac{\#\text{documents dislike}}{\text{total number of documents}}$$

$$\mathbf{v_{NB} = \underset{v \in \{like, dislike\}}{argmax}\ P(v) \prod_i P(x_i = word_i \mid v)}$$

How to compute probabilities?

- Second, compute $P(x_i = word_i \mid v)$ for each word:

$$P(word_k \mid v) = \frac{\#(\text{docs. v in training where } word_k \text{ apears})}{\#(\text{documents v})} = \frac{n_k}{n}$$

- Number of parameters to estimate is not too large: 10.000 words and two classes (so about 20.000)

- Problem:

$$P(word_k \mid v) = \frac{\#(\text{docs. v del training on } word_k \text{ apareix})}{\#(\text{documents v})} = \frac{n_k}{n}$$

  - When n_k is low, not an accurate probability
  - when n_k is 0 for $word_k$ for one class v, then any document with that word will never be assigned to v (independent of other appearing words)

- Solution: More robust computation of probabilities (Laplace *smoothing*)

$$P(word_k \mid v) = \frac{n_k + mp}{n + m}$$

- Where:
  - n_k is # of documents of class v in which word *k* appear
  - n is # of documents with label *v*
  - *p* it's a likelihood estimation of "a priori" $P(x_k|v)$ (f.i., uniform distribution)
  - m is the number of labels

Smoothing: $P(x_k \mid v) = \frac{n_k + mp}{n + m}$

More common "a priori" uniform distribution:

1. When two classes: p=1/2, m=2 (Laplace Rule)

$$P(x_k \mid v) = \frac{n_k + 1}{n + 2}$$

2. Generic case (***c*** classes): p = 1/***c***, m=***c***

$$P(x_k \mid v) = \frac{n_k + 1}{n + c}$$

- Assume each class follows a normal distribution for each variable

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

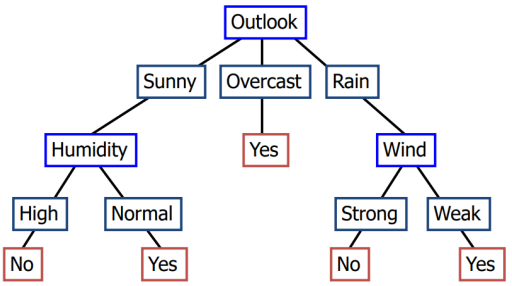
- For instance 73 is average of feature temp. for class x, and std=26.2, we compute conditional prob in the following way:

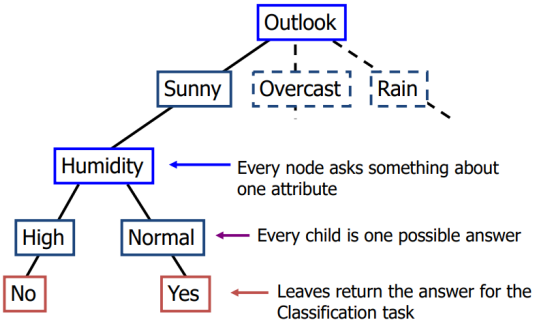
$$p(temperature = 66 \mid x) = \frac{1}{\sqrt{2\pi}6.2} e^{\frac{(66-73)^2}{26.2^2}} = 0.034$$

Tema 4.2: Decision Trees

1/47

Day	Outlook	Temp.	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cold	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No



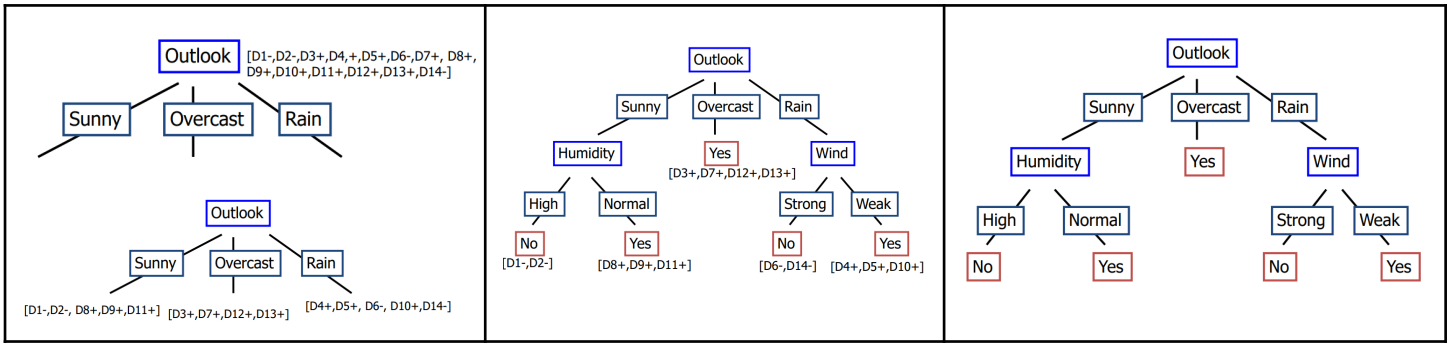


(Outlook=Sunny $\wedge$ Humidity=Normal)
$\vee$ (Outlook=Overcast)
$\vee$ (Outlook=Rain $\wedge$ Wind=Weak)

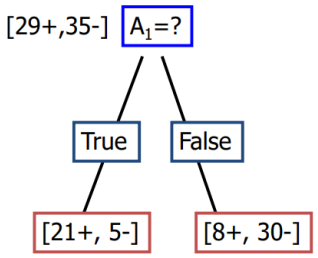
Amb un training set-molts possibles arbres
Buscar l'arbre més simple- serà el més curt

Top-Down: Induction de Decision Trees

1. Posar node del training exemple com a root
2. A<- millor **característica** per el node seleccionat
3. Per cada valor de la característica A, crear nou fill
4. Separar els exemples en nodes depenen dels fills depenen dels valors de A
5. Si hi han **fulles** nodes amb barreja de negatius i positius exemples-> retornar pas 2 fulla passa a node
6. Posar a cada fulla una **etiqueta** amb l'exemple de la fulla i retonaho



Aconseguir el arbre més curt ⇒ separar lo més ràpid possible
+exemples de -exemples
Preguntes **molt discriminatives**



Teoria de la informació:
Una resposta ens dóna una informació proporcional a la resposta no predectible

Ex: monedes- cara o creu

$$E(X) = -\sum_{i}^{n} p_i \log(p_i)$$

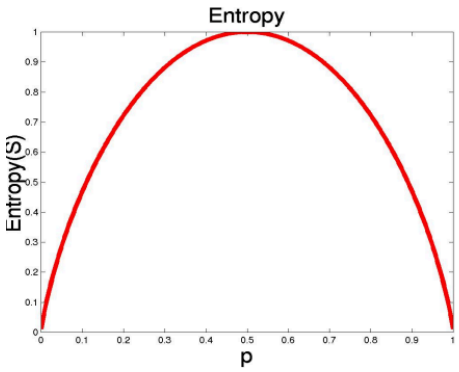
pi probabilitat de la resposta i
n nombre de possibles respostes

Informació guanyada (Information Gain) serà:

$$IG(\mathcal{X}, A) = E(\mathcal{X}) - E(\mathcal{X}|A)$$

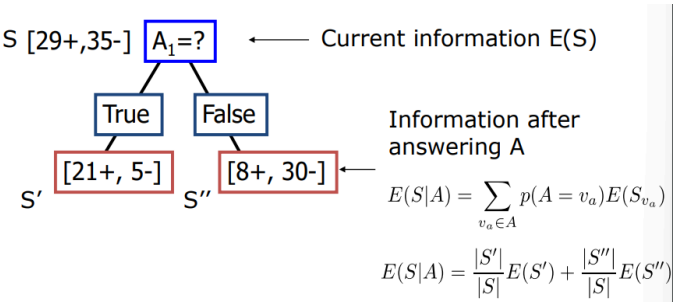
Resposta binària:

$E(A) = -p_+ \log_2 p_+ - p_- \log_2 p_- =$
$-p_+ \log_2 p_+ - (1-p_+)\log_2 (1-p_+)$



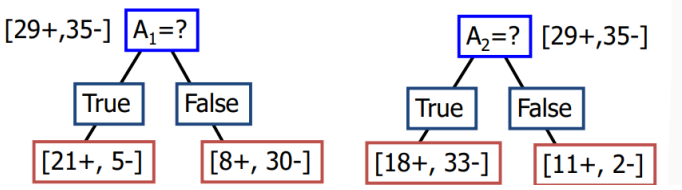
- For DTs, we assume:
  - p_+ the proportion of + examples in S
  - p_- the proportion of – examples in S
- E(S) measures the "mixing" of examples in S
  $E(S) = -p_+ \log_2 p_+ - p_- \log_2 p_-$

Entropia (Entropy):
24/47
E(s) - Entropia de S- mesura la **necessitat de bits** per comprimir la informació de la distribució de +/- exemples considerant que es aleatori.
Mesura del contingut del a informació de la teoria de la informació
IG- la diferencia entre la informació abans i després de la resposta SUMATORI dels casos

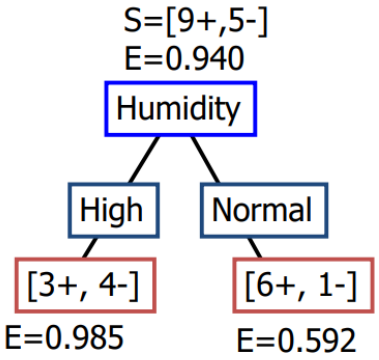


$E([29+,35-]) = -29/64 \log_2 29/64 - 35/64 \log_2 35/64 = 0.99$

$IG(S,A)=E(S) - \sum_{v \in values(A)} |S_v|/|S| \; E(S_v)$



Com triar el millor atribut? → el que tingui **la informació guanyada més alta**



Trobar el num de dades a treballar → de +9 a -5 ⇒ 14

$E(Humitat) = -9/14 * log2(9/14) + -5/14\, log2\,(5/14)$

E(humitat)=0.940

3 a -4 ⇒7

$E(H, high) = -3/7 * log2(3/7) - 4/7\, log2\,(4/7)$

E(h,High)=0.985

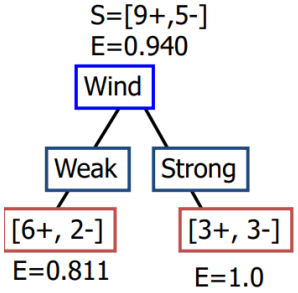
..

E(H,normal)=0.592

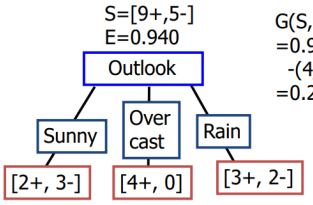
IG(S,Humitat) = E(humitat) - 7/14 * E(H,high) - 7/14 * E(H,normal)

G(S,Humitat)=0.940-(7/14)*0.985 – (7/14)*0.592 =**0.151**

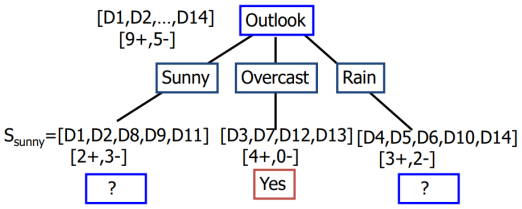
Calcular per cada atribut:



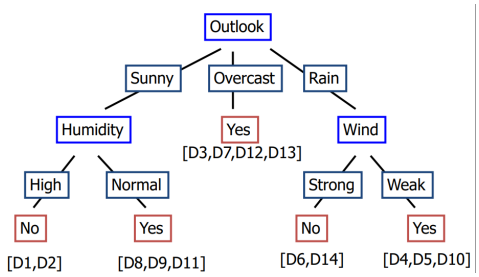
G(S,Wind)
=0.940-(8/14)*0.811
– (6/14)*1.0
=0.048



G(S,Outlook)
=0.940-(5/14)*0.971
-(4/14)*0.0 – (5/14)*0.0971
=0.247



Gain(S_{sunny} , Humidity)=0.970-(3/5)0.0 – 2/5(0.0) = 0.970
Gain(S_{sunny} , Temp.)=0.970-(2/5)0.0 –2/5(1.0)-(1/5)0.0 = 0.570
Gain(S_{sunny} , Wind)=0.970= -(2/5)1.0 – 3/5(0.918) = 0.019



Humidity -> Temp(només te una opcio) -> Wind

De més a menys valor de IGain

<u>Overfitting:</u>

33/47

Possible problema →Si no hi han exemples en un valor quan expandim un node

No generar la branca - anar a la branca més popular

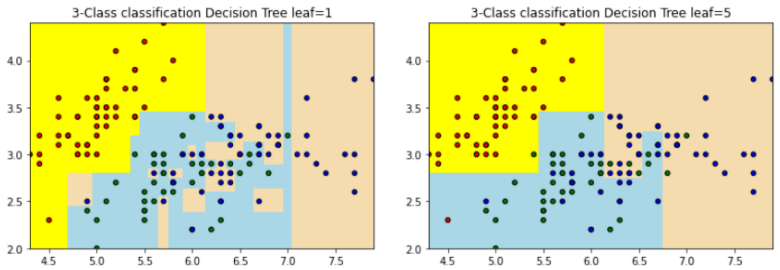
Overfit les dades donades q podem separar amb exemples fins que arribem a una fulla on totes les prediccions son correctes

Cada vegada que descendem per l'arbre emportar menys i menys dades

Regular el learning controlant paràmetres:

- Min numeros de exemples en una fulla
- número de nodes/fulles en un arbre
- Max profunditat de branques en un arbre
- Arreglar threshold per la heurística

Més fulles més classificat →



Alternativa es fer Overfitting i **Prune** els resultats estimant la generalització d'errors per les fulles -> eliminant nodes amb un error més gran a no tenir aquell node (després de prunar)

Hi haurà un biased a les característiques amb molts valors diferents en variables categòriques

Variables categòriques:

- IG té preferència als atributs amb més número de valors
  Tenen més possibles respostes → ordenar més les dades
- Implementació practica assumir atributs categòrics com a **binaris**
- Atributs amb més de 2 valors transformats **one-hot encoding**

Atributs amb valors numèrics:

Donat un continus atribut A, considerar tots diferents divisions

$$[A[x] < v_i] \text{ versus } [A[x] >= v_i]$$

- Començar amb seqüència de valors A= {v1,v2,..vn}
- Considerar cada valor té un **threshold** per **split**
  - Per cada split calcular la **information gain** (n-1 possibles splits)
- Fer split amb el IG més alt a un **test booleà** per el node de dades que es faci split
- L'atribut A no es pot eliminar de futurs splits
- <u>Inconvenients</u>: puja molt el temps de computació

Ex: atribut de temperatura

64	65	68	69	70	71	72	72	75	75	80	81	83	85
Yes	No	Yes	Yes	Yes	No	No	Yes	Yes	Yes	No	Yes	Yes	No

Posar split al mig dels valors

- E.g.: temperature < 71.5: yes/4, no/2
  temperature ≥ 71.5: yes/5, no/3

Evaluar tots els split points en una passada

- Info([4,2],[5,3]) = 6/14 info([4,2]) + 8/14 info([5,3]) = 0.939 bits

Per agilitzar procés -> Entropia només cal evaluar punts entre les diferents classes

value	64	65	68	69	70	71	72	72	75	75	80	81	83	85
class	Yes	No	Yes	Yes	Yes	No	No	Yes	Yes	Yes	No	Yes	Yes	No

Potencials **optimal breakpoint**s -> entre valors de mateixa classe que **no** poden ser optims

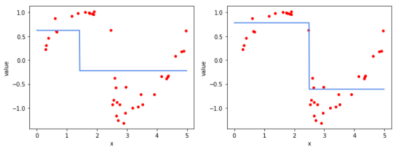
<u>Arbre de decisió per regressió:</u>

Similar classificació- minimitza el sumatori d'errors relatius al valor target mitjà de cada partició inducida

Find pair (v^*, s^*) where v^* is an input feature, and $s^* \in R$ s.t.

$$v^*, s^* := \arg\min_{v,s} \left[SSE\left(S_{v \le s}\right) + SSE\left(S_{v > s}\right)\right]$$

where $\mathrm{SSE}(S) := \sum_{i:(\mathbf{x}_i, y_i) \in S} \left(y_i - \mathrm{avg}(S)\right)^2$



- <u>Avantatges</u>:

Computational complexity baixa, només atributs rellevants s'usen, facil interpretar

- <u>Inconvenients</u>:

Molta variança, qualsevol canvi pot canviar el model, sensibles al *noise*, aproximen els conceptes per particions que son paral·lels als attributs, poden retornar models complexes

Tema 5.1 Boosting

1/69

Multiclassificadors,meta learners, ensemble learners

- Combina varis *febles* learners => fort learner
- ensamble aplicat a un sol tipus de feble learner- mateix algoritme diferents dades/param.

1) Aconsegueixen una alta precisió (accuracy)a la pràctica - canviar computation time per debilitat del calssifier
2) Combina forces de diferents *classifier buliders*
3) Ajuda a evitar overfitting

Combinar learners febles:

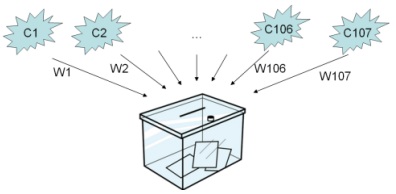
- Voting: cada learner feble vota - vots es combinen
- Experts s'abstenen (a votar) - només els learners febles conten si el seu expert s'abstent

Esquemes de votació:

10/69

Combinem les vots:

- Segons la majoria de vots
- El **pes** depèn dels errors s (1 − ei ? 1/ei ? exp(−ei)? . . .)
- El **pes** depèn de la **confiança (confidence)**
- maximitzar la diversitat



Stacking:

16/69

Meta-learner que aprèn el **pes** dels learners més febles

- Dataset instàncies de (x,y)
- Transformar data set per tenir instàncies (x,c1(x),...cN(x),y)
- Entrena el metaclassificador M amb el dataset enriquit

Normalment no es dona x a M, sinó només els vots

Clasificador lineal

Pot simular casi tots els esquemes de votació

Weighted majority:

```
initialize classifiers c1...cN with weight wi = 1/N each;
for each example x in sequence do
    collect predictions c1(x)...cN(x);
    prediction(x) = sign[ w1*c1(x)+...+wN*cN(x))-1/2 ]
    get true label y for x;
    for each i=1..N,
        if (ci(x) != y) then wi = wi/2;
    renormalize weights to sum 1;
```

19/69

El pes depèn exponencialment del error

Millor temps O log(N)

Normalment millor classificador, quan els classificadors no estan relacionats

Bo per prediccions online amb molts classificadors

ex: 1 classificador =1 característica

Bagging and Random Forest:

31/69

1. Donat dataset de S amb N exemples etiquetats amb A atributs:
2. Crea N barring replicas de S: S1,..Sn
    a. Si= dibuixa N exemples per S amb replacement
3. Usar N replicas per construir **N febles learners C1,..CN**
4. Prediir usant la **majoria del vot de Ci's**

Bagging:

Majority voting amb pocs samples o només un classifier builder

Millora inestables febles learners- alta variança

Potser empitjora estables febles learners- baixa variança

Útil en Decision Trees i Perceptrons - alta variança o sensibilitat al training set data

Example of building training sets:

Original:	1	2	3	4	5	6	7	8
Training Set1:	2	7	8	3	7	6	3	1
Training Set2:	7	8	5	6	4	2	7	1
Training Set3:	3	6	2	7	5	6	2	2
Training Set4:	4	5	1	4	6	4	3	8

Random Forests:

1. Paràmetres k i a;
2. Dataset S amb N exemples etiquetats amb A atributs:
3. Crea k **bagging replicas** de S: S1,..Sk
4. Usar k replicas per crear k **random trees T1,...Tk**
    a. Cada node divideix, aleatòriament a <=A atributs, triar **millor de les a**
    b. Creix cada arbre el **màxima profunditat** possible (no pruning)
5. Predir usant la **majoria de vots de Ti's**

pes feble learner vs variança feble learner

- Més atributs millora pes- overfits més
- Més arbres t disminueix variança - overfits ments

Màxim performance en moltes tasques

Boosting:

38/69

Intenta reduir variança de classificadors base creant diferents **bootstrapping datasets**

Intenta activament millorar la precisió de febles classificadors

com? → Entrena una seqüencia de **classificadors especialitzats** basats en **errors previs**

Adaptament.sequencialment crea classificadors

Classificadors i instàncies tenen pes variable

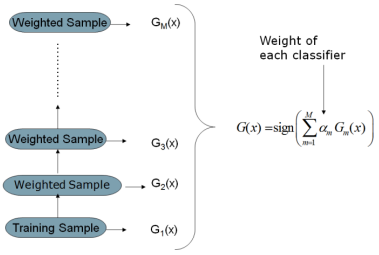
Augmenta el pes de instàncies classificades incorrectament

La **etiqueta final** es *weighting voting* de seqüència de classificadors

Treballa sobre qualsevol feble learner - definit amb qualsevol learning mechanism que funciona millor que casualitat (precisió > 0.5 2casos equivalentment probables)

- Only two classes
- Output: $y \in \{-1, 1\}$
- Examples: X
- Weak Classifier: G(X)
- Error de training (err_{train})

$$err_{train} = \frac{1}{N} \sum_{i=1}^{N} I(y_i \neq G(x_i))$$



Adaboost algorithm:

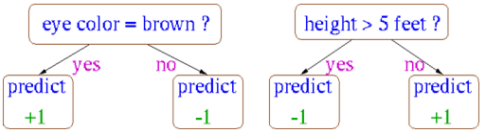
Set weight of all examples to $1/n$ For t=1:L S_t = training set using weights for each example Learn $G_t(S_t)$ Compute err_t for G_t Compute $\alpha_t = \frac{1}{2}\ln\left(\frac{1-err_t}{err_t}\right)$ Compute new weights $w_i \leftarrow \frac{w_i}{Z_t} \cdot e^{-[\alpha_t \cdot y_i \cdot G_t(x_i)]}$ Return classifier: $G(x) = \text{sign}\left(\sum_{t=1}^{L} \alpha_t G_t(x)\right)$	$\alpha_m = \frac{1}{2}\ln\left(\frac{1-err_m}{err_m}\right) > 0$ $w_i \leftarrow \frac{w_i}{Z_m} \cdot e^{-[\alpha_m \cdot y_i \cdot G(x_i)]}$ $G(x) = \text{sign}\left(\sum_{m=1}^{L} \alpha_m G_m(x)\right)$	$\alpha_m = \frac{1}{2}\ln\left(\frac{1-err_m}{err_m}\right) > 0$ $w_i \leftarrow \frac{w_i}{Z_m} \cdot \begin{cases} e^{-\alpha_t} & \text{if } y_i = G_t(x_i) \\ e^{\alpha_t} & \text{if } y_i \neq G_t(x_i) \end{cases}$ $G(x) = \text{sign}\left(\sum_{m=1}^{L} \alpha_m G_m(x)\right)$

AdaBoost 3:

Theorem. *Suppose that the error of classifier h_t is $1/2 - \gamma_t$, $t = 1..T$. Then the error of the combination H of $h_1, \ldots h_T$ is at most*

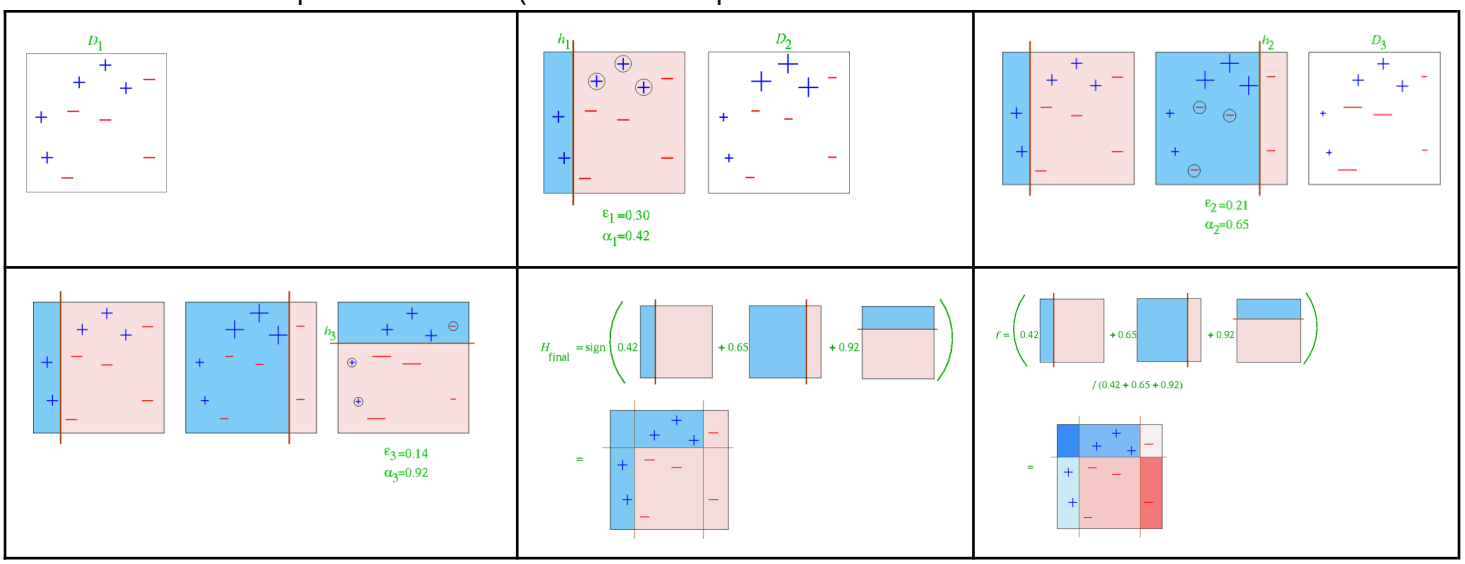
$$\exp\left(-\sum_{t=1}^{T} \gamma_t^2\right)$$

Note: It tends to 0 if we can guarantee $\gamma_i \geq \gamma$ for fixed γ



Exemple:

Decision stumps- feble learner (decision trees pruned 1 nivell



Boosting vs Bagging:

Boosting dona una millora en la precisió que en Bagging

Boosting s'equivoca algunes vegades - si hi ha *noisy data*

Baggin cosisteixament dona una millora

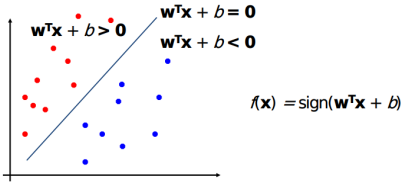
1) Estadisticament: no podem confiar a un sol classificador, reduim la variança
2) Computational reasons: feble classificador pot estar parat en una *local minima*. començant amb diferents training data sets podem trobar millors solucions
3) Representational reasons: Combinació de classificadors retorna una solució fora de la hyipotesis inicial, s'adapten un millor al problema

69/69

Tema 5.2 Support Vector Machines

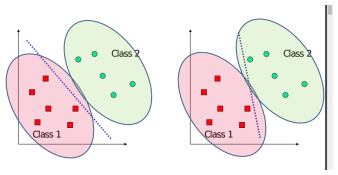
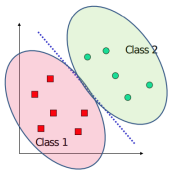
1/48

Large-margin linear classifier:



Linear Separable:

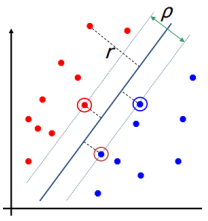
Linear separators: classificador binari com tasca de separar classes en un espai
Com considerar un bon separador lineal?
l'algorisme de percepció es pot usar per trobar un límit

Mals límits de decisió	Millors limits de decisió
	

Marge de classificació:
Distància del exemple Xi el separador es

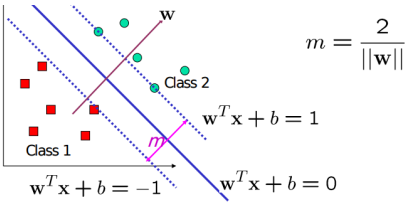
$$r = \frac{\mathbf{w}^T\mathbf{x}_i + b}{\|\mathbf{w}\|}$$

Creen els **support vectors**
El marge p del separador és la distància entre els support vectors



El límit de decisió (decision boundary) hauria d'estar el màxim lluny de les dades de les dues classes.

- Maximitzar el marge m m=2/||w||
- La distancia entre origen i la lína wtx=k i k /||w||



Maximitzar el marge bona decisió segons intuïció i teoria PAC
→ implica que només els SVM importen i training examples son ingnorables

Trobar el límit de decisió:

- Let $\{x_1, \ldots, x_n\}$ be our data set and let $y_i \in \{1,-1\}$ be the class label of x_i
- The decision boundary should classify all points correctly

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \qquad \forall i$$

- The decision boundary can be found by solving the following constrained optimization problem

$$\text{Minimize } \frac{1}{2}||\mathbf{w}||^2$$
$$\text{subject to } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \qquad \forall i$$

- This is a constrained optimization problem. Solving it requires some new tools
  - Feel free to ignore the following several slides; what is important is the constrained optimization problem above

optimització restringida

- Suppose we want to: minimize f(**x**) subject to g(**x**) = 0
- A necessary condition for $\mathbf{x}_0$ to be a solution:

$$\begin{cases} \frac{\partial}{\partial \mathbf{x}}\left(f(\mathbf{x}) + \alpha g(\mathbf{x})\right)\Big|_{\mathbf{x}=\mathbf{x}_0} = 0 \\ g(\mathbf{x}) = 0 \end{cases}$$

- α: the Lagrange multiplier
- For multiple constraints $g_i(\mathbf{x}) = 0$, i=1, ..., m, we need a Lagrange multiplier α_i for each of the constraints

$$\begin{cases} \frac{\partial}{\partial \mathbf{x}}\left(f(\mathbf{x}) + \sum_{i=1}^{n} \alpha_i g_i(\mathbf{x})\right)\Big|_{\mathbf{x}=\mathbf{x}_0} = 0 \\ g_i(\mathbf{x}) = 0 \qquad \text{for } i = 1, \ldots, m \end{cases}$$

optimització restringida

- The case for inequality constraint $g_i(\mathbf{x}) \leq 0$ is similar, except that the Lagrange multiplier α_i should be positive
- If $\mathbf{x}_0$ is a solution to the constrained optimization problem

$$\min_{\mathbf{x}} f(\mathbf{x}) \quad \text{subject to} \quad g_i(\mathbf{x}) \leq 0 \quad \text{for } i = 1, \ldots, m$$

- There must exist $\alpha_i \geq 0$ for i=1, ..., m such that $\mathbf{x}_0$ satisfy

$$\begin{cases} \frac{\partial}{\partial \mathbf{x}}\left(f(\mathbf{x}) + \sum_i \alpha_i g_i(\mathbf{x})\right)\Big|_{\mathbf{x}=jx_0} = 0 \\ g_i(\mathbf{x}) \leq 0 \qquad \text{for } i = 1, \ldots, m \end{cases}$$

- The function $f(\mathbf{x}) + \sum \alpha_i g_i(\mathbf{x})$ is also known as the Lagrangrian; we want to set its gradient to **0**

problema original

$$\text{Minimize } \frac{1}{2}||\mathbf{w}||^2$$

$$\text{subject to } 1 - y_i(\mathbf{w}^T\mathbf{x}_i + b) \leq 0 \qquad \text{for } i = 1, \ldots, n$$

- The Lagrangian is

$$\mathcal{L} = \frac{1}{2}\mathbf{w}^T\mathbf{w} + \sum_{i=1}^{n} \alpha_i \left(1 - y_i(\mathbf{w}^T\mathbf{x}_i + b)\right)$$

  - Note that $||\mathbf{w}||^2 = \mathbf{w}^T\mathbf{w}$
- Setting the gradient of $\mathcal{L}$ w.r.t. **w** and b to zero, we have

$$\mathbf{w} + \sum_{i=1}^{n} \alpha_i(-y_i)\mathbf{x}_i = \mathbf{0} \quad \Rightarrow \quad \mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

Problema dual

- If we substitute $\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$, we have $\mathcal{L}$

$$\mathcal{L} = \frac{1}{2}\sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i^T \sum_{j=1}^{n} \alpha_j y_j \mathbf{x}_j + \sum_{i=1}^{n} \alpha_i \left(1 - y_i(\sum_{j=1}^{n} \alpha_j y_j \mathbf{x}_j^T \mathbf{x}_i + b)\right)$$

$$= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j + \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \alpha_i y_i \sum_{j=1}^{n} \alpha_j y_j \mathbf{x}_j^T \mathbf{x}_i - b\sum_{i=1}^{n} \alpha_i y_i$$

$$= -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j + \sum_{i=1}^{n} \alpha_i$$

- Note that $\sum_{i=1}^{n} \alpha_i y_i = 0$
- This is a function of α_i only

- The new objective function is in terms of α_i only
- It is known as the dual problem: if we know **w**, we know all α_i; if we know all α_i, we know **w**
- The original problem is known as the primal problem
- The objective function of the dual problem needs to be maximized!
- The dual problem is therefore:

$$\max. \ W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1,j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

$$\text{subject to } \alpha_i \geq 0, \quad \sum_{i=1}^{n} \alpha_i y_i = 0$$

Properties of α_i when we introduce the Lagrange multipliers

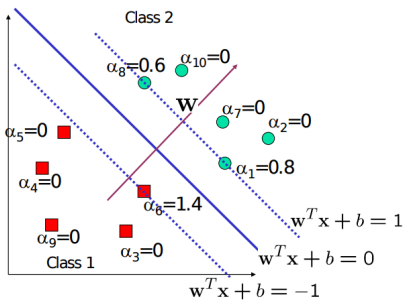
The result when we differentiate the original Lagrangian w.r.t. b

$$\max. \ W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1,j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

$$\text{subject to } \alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i y_i = 0$$

- This is a quadratic programming (QP) problem
  - A global maximum of α_i can always be found
- **w** can be recovered by $\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$

Interpretació geomètrica:



Característiques de la solució:

- Many of the α_i are zero
  - **w** is a linear combination of a small number of data points
  - This "sparse" representation can be viewed as data compression as in the construction of knn classifier
- $\mathbf{x}_i$ with non-zero α_i are called support vectors (SV)
  - The decision boundary is determined only by the SV
  - Let t_j (j=1, ..., s) be the indices of the s support vectors. We can write $\mathbf{w} = \sum_{j=1}^{s} \alpha_{t_j} y_{t_j} \mathbf{x}_{t_j}$
- For testing with a new data **z**
  - Compute $\mathbf{w}^T\mathbf{z} + b = \sum_{j=1}^{s} \alpha_{t_j} y_{t_j} (\mathbf{x}_{t_j}^T \mathbf{z}) + b$ classify **z** as class 1 if the sum is positive, and class 2 otherwise
  - Note: **w** need not be formed explicitly

programació:

- Many approaches have been proposed
  - Loqo, cplex, etc. (see http://www.numerical.rl.ac.uk/qp/qp.html)
- Most are "interior-point" methods
  - Start with an initial solution that can violate the constraints
  - Improve this solution by optimizing the objective function and/or reducing the amount of constraint violation
- For SVM, sequential minimal optimization (SMO) seems to be the most popular
  - A QP with two variables is trivial to solve
  - Each iteration of SMO picks a pair of (α_i,α_j) and solve the QP with these two variables; repeat until convergence
- In practice, we can just regard the QP solver as a "black-box" without bothering how it works

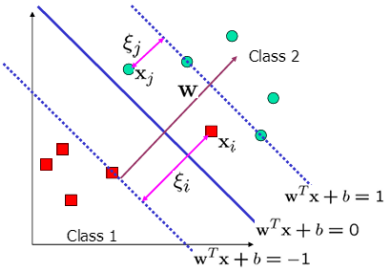
Nonlinear Separable Datasets: Soft-Margin SVM

25/48

Algunes vegades les dades no son linearment separables→ o no volem separar-les perfectament

Slack variables:ε poden ser afegides per acceptar una errònia classificació per dades difícils o noisy, tenint així un marge **soft**

- Acceptem un cert error a la classificació: basat en el output segons el punt i la recta lineal que tinguem creada
- Ha de ser diferent a 0 per dades mal classificades



Limitacions:

If we minimize $\sum_i \xi_i$, ξ_i can be computed by

$$\begin{cases} \mathbf{w}^T\mathbf{x}_i + b \geq 1 - \xi_i & y_i = 1 \\ \mathbf{w}^T\mathbf{x}_i + b \leq -1 + \xi_i & y_i = -1 \\ \xi_i \geq 0 & \forall i \end{cases}$$

- ξ_i are "slack variables" in optimization
- Note that ξ_i=0 if there is no error for $\mathbf{x}_i$
- ξ_i is an upper bound of the number of errors

We want to minimize $\frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^n \xi_i$

- C : tradeoff parameter between error and margin

The optimization problem becomes

$$\text{Minimize } \tfrac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^n \xi_i$$
$$\text{subject to } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0$$

The dual of this new constrained optimization problem is

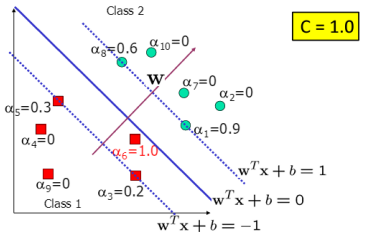
$$\text{max. } W(\boldsymbol{\alpha}) = \sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i=1,j=1}^n \alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j$$

$$\text{subject to } C \geq \alpha_i \geq 0, \sum_{i=1}^n \alpha_i y_i = 0$$

w is recovered as $\mathbf{w} = \sum_{j=1}^s \alpha_{t_j} y_{t_j} \mathbf{x}_{t_j}$

This is very similar to the optimization problem in the linear separable case, except that there is an upper bound C on α_i now

Once again, a QP solver can be used to find α_i



Importància dels supports:

- Els supports son punts a la frontera entre les diferents classes **(supports +errors)**
- Solucó pot ser estructurada només amb els supports
- Nombre supports més petita que la dimensió donada
- Nombre supports és limit superior de **Leave-one-out-error**

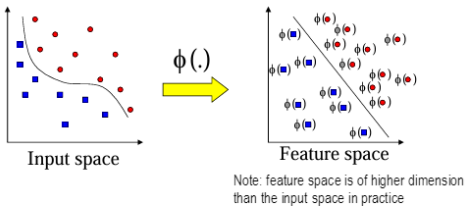
Non-linear classifier: Kernel trick:

35/48

Que fer si es una línea de decisió **no linear?**

→ Transformar Xi a una dimensió més alta facilitar classificació

- *input space:* l'espai on el punt xi esta
- *feature space:* l'espai on Φ(Xi) esta després de transformació



<u>Transformar dades en una dimensió més:</u>

El espai original *feature space* pot ser mapejat per una característica de *feature space* on les dades del training set son separables

Computació del *feature space* costós →**kernel trick**

$$\text{max. } W(\boldsymbol{\alpha}) = \sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i=1,j=1}^n \alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j$$

$$\text{subject to } C \geq \alpha_i \geq 0, \sum_{i=1}^n \alpha_i y_i = 0$$

The data points only appear as inner product

Tipus de Kernel:

- All kernels has the following form

$$K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j) = N^T N$$

- Any matrix that can be decomposed as $N^T N$ is called as **symmetric, positive definite matrix (sdp)**
- **Any function** *K(x,z)* that creates a **symmetric, positive definite matrix is a valid kernel** (= an inner product in some space)
- ...even when we don't know projection function $\phi(.)$
- This is the case of the RBF function

- Polynomial kernel **with degree *d***

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T \mathbf{y} + 1)^d$$

- Radial basis function kernel **with width σ**

$$K(\mathbf{x}, \mathbf{y}) = \exp(-||\mathbf{x} - \mathbf{y}||^2/(2\sigma^2))$$

Com triar funció del kernel: crea la matriu del kernel→ agrupa totes les dades
A la pràctica triar entre: **polynomial kernel o RBF kernel** provar inicialment

Passos classificació:

1) Preparar matriu de dades **numeric+normalització**
2) Seleccionar kernel a utilitzar
3) Seleccionar paràmetre del kernel i **valor de C**
4) Exectuar *training algorithm* i obtenir α_i
5) Les dades no vistes classificades per α_i **i els support vectors**

<u>Avantatges SVM:</u>

- Training data és relativament fàcil
- Escala bé les dades amb alta dimensionalitat
- Intercanvi entre complexitat i error pot ser controlada
- Dades no tradicionals (strings o arbres) poden ser input SVM

<u>Inconvenients:</u>

- Necessita triar correctament una funció kernel

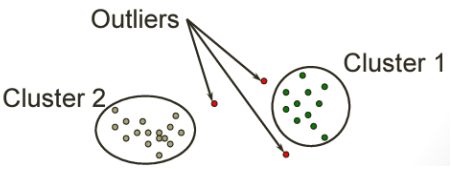
Tema 6: Clustering

1/115

Clustering:

Grup de dades en **clusters** :

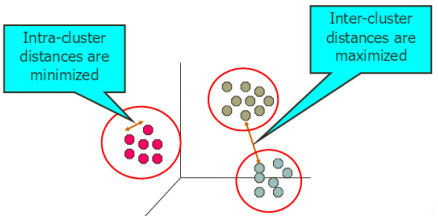
similar un entre l'altre dins del mateix cluster
no similars amb els objectes en altres clusters
Unsupervised learning no es predefineixen les classes



Clusters útils → descobreix nous coneixements per les dades, contrasta amb *supervised classification (*etiquetes conegudes*)*, útil per resumir grans datasets

Distàncies i Mesures:

Un **bon mètode de cluster,** produirà bona qualitat de clusters:
high intra-class similarity, low inter-class similarity



La **qualitat** de clustering

- depèn de la mesura de la similitud i la seva implementació
- depèn de l'habilitat de descobrir alguns o tots els patrons ocults

Mesura de similitud:

termes de funció de distància *d(i,j), interval scaled, boolean,categorical, ordinal and ratio*
El pes hauria d'estar associat a diff. variables basades en aplicació i semàntica de dades
qualitat de funció mesures la **bona feina** del cluster
"prou similar" o "prou bo" difícil de definir-> molt subjectiu

- The word "***similarity***" in clustering means that the value of **s(x, x')** is large when x and x' are two similar samples; the value of s(x, x') is small when x and x' are not similar.
- Similarity measure *s* is symmetric:

$$s(x, x') = s(x', x), \quad \forall x, x' \in X$$

- Similarity measure is normalized:

$$0 <= s(x, x') <= 1, \quad \forall x, x' \in X$$

Dissimilarity measure is denoted by **d(x, x')**, $\forall x, x' \in X$, and it is frequently called a **distance**:

- $d(x, x') >= 0, \quad \forall x, x' \in X$
- $d(x, x') = d(x', x), \quad \forall x, x' \in X$
- $d(x, x'') <= d(x, x') + d(x', x''), \quad \forall x, x', x'' \in X$ (triangular inequality)

1. Euclidean distance in m-dimensional feature space:

$$d_2(x_i, x_j) = \left(\sum_{k=1}^{m} (x_{ik} - x_{jk})^2 \right)^{1/2}$$

2. L_1 metric or **city block distance**:

$$d_1(xi, xj) = \sum_{k=1}^{m} |x_{ik} - x_{jk}|$$

3. Minkowski metric (includes the Euclidean distance and city block distance as special cases):

$$d_p(x_i, x_j) = \left(\sum_{k=1}^{m} (x_{ik} - x_{jk})^p \right)^{1/p}$$

4. Cosine-correlation:

$$\mathrm{scos}(x_i, x_j) = \frac{\sum_{k=1}^{m} (x_{ik}\ x_{jk})}{\left(\sum_{k=1}^{m} x_{ik}^2 \times \sum_{k=1}^{m} x_{jk}^2 \right)^{1/2}}$$

$$\mathrm{scos}(x_i, x_j) = 1 \Leftrightarrow \forall i, j \text{ and } \lambda > 0 \text{ where } x_i = \lambda \cdot x_j$$

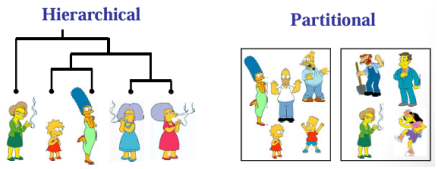
$$\mathrm{scos}(x_i, x_j) = -1 \Leftrightarrow \forall i, j \text{ and } \lambda < 0 \text{ where } x_i = \lambda \cdot x_j$$

Algorismes:

23/115

Dos tipus algorismes:

- ***Partitional algorithms***:Construeix varies particions i avaluar per algun criteri
- ***Hierarchical algorithms***: Construeix una descomposició jerárquica dels objectes utilitzant algun criteri

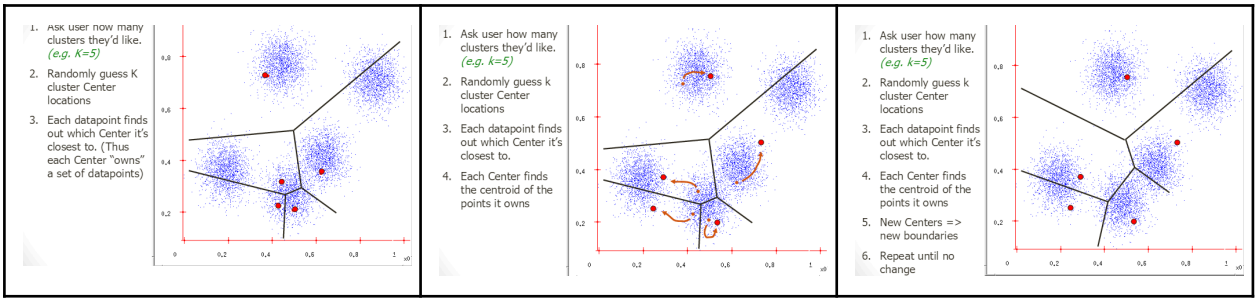


No hi ha un millor algorisme de cluster per TOTS els problemes, els outputs només donaran suggerencies o confirmar hipótesi, però mai provar hipòtesi sobre l'organització natural de els dades

Cluster K-means:

Partitional algorithm. Cada Cluster és associat amb un **centroid** (punt central)
Cada punt assignat al cluster amb el centroid més aprop
Nombre de K clusters ha destar especificat
Algorisme:

```
Select K points as the initial centroids.
repeat
    Form K clusters by assigning all points to the closest centroid.
    Recompute the centroid of each cluster.
until The centroids don't change
```



Els centroids inicials son triats aleatòriament, normalment son la mitjana dels punts del cluster
La proximitat mesurada per: *Euclidean distance, cosine similarity, correlation …*
Kmeans *convergiran* per les mesures de similitud donades anteriorment.
La major convergència es farà a les primeres iteracions
Complexitat: O(n * K * I * d) n = number of points, K = number of clusters,
I = number of iterations, d = number of attributes

<u>Possible problema amb seleccionar punts inicials:</u>
Si hi ha K clusters, probabilitat seleccionar un centroid en cada cluster és petita
Si clusters son del mateix tamany:

$$P = \frac{\text{number of ways to select one centroid from each cluster}}{\text{number of ways to select } K \text{ centroids}} = \frac{K!n^K}{(Kn)^K} = \frac{K!}{K^K}$$

<u>Evaluar K-means clusters:</u>

42/115
Més comú **Sum de L'error al quadrat (SUM of Squared Error SSE)**
Per cada punt, l'error és la distància del cluster més proper
SSE arrel dels errors i la suma d'aquests

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(m_i, x)$$

- x is a data point in cluster C_i and m_i is the representative point for cluster C_i
  - can show that m_i corresponds to the center (mean) of the cluster

→ Triar el que té menys error

Poder reduir valor SSE, augmentar K→ bon clustering menor k, millor SSE que pitjor clustering amb major k
<u>Possibles solucions punts inicials:</u>

- Múltiples execucions: probablement dolent
- *Hierarchical clustering* per determinar centroids inicials

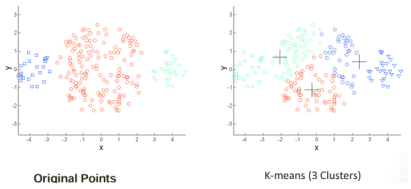
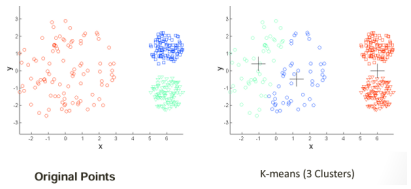
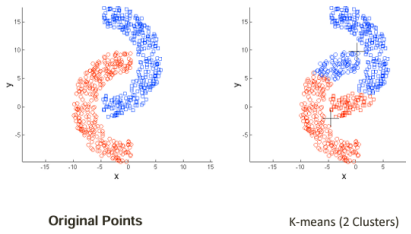
- Seleccionar més de k centroids inicials i seleccionar els centroids més separats

<u>Preprocessament i Postprocessament:</u>

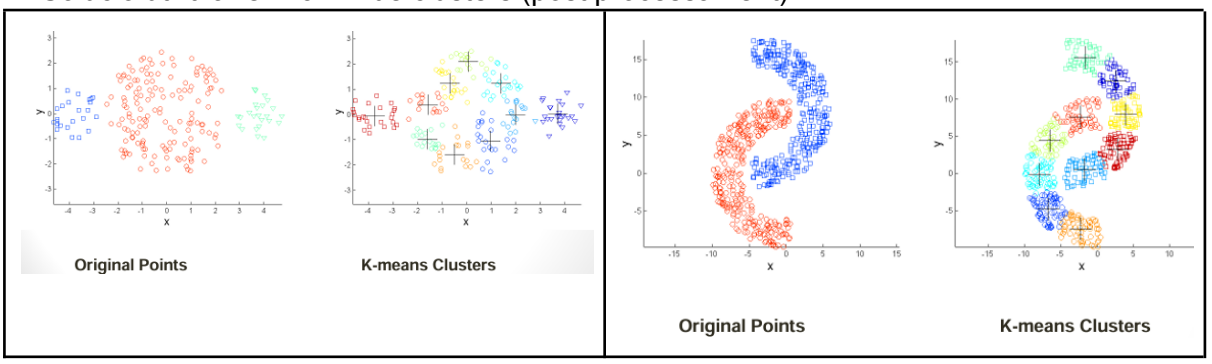
Preprocessament:
- Normalitzar les dades
- Eliminar outliers

Postprocessament:
- Eliminar petits clusters - possibles outliers
- Ajuntar clusters molt aprop i petit valor SSE
  (utilitzar els punts durant el processament del cluster)

Limitacions:
- K Means té problemes quan els clústers tenen diferències de:
  - tamany
  - densitat
  - formes no globular
- Problemes si dades contenen outliers

diferents tamanys	diferents densitats	formes no globulars
		

→ Solució utilitzar el màxim de clusters (post processament)

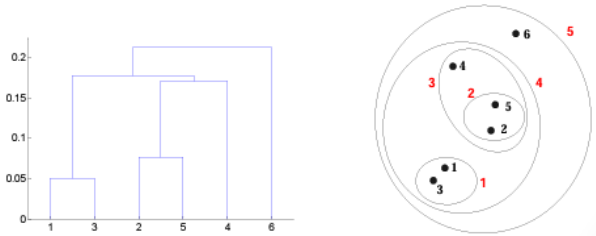


<u>Hierarchical clustering:</u>

59/115

Produeix un set de *nested clusters* organitzats en un **arbre jeràrquic**

Pot ser visualitzat en un **dendrogram:** recorda la seqüència de fusions o divisions



No s'ha d'assumir cap nombre particular de clusters
Poden correspondre a taxonomies significatives(*animal kingdom phylogeny reconstruction..*)
Dos tipus de hirechical clustering:

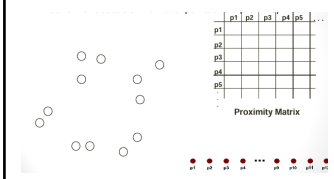
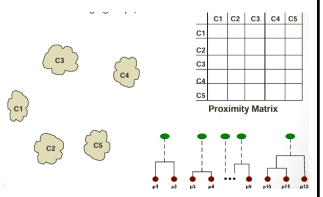
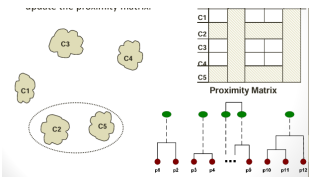
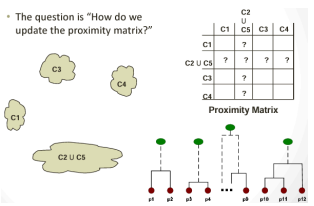
- **Agglomerative:**
    - Comença amb punts com clusters individuals
    - Cada iteració fusiona el parell de clústers que estan més aprop fins que només hi hagi un sol cluster (o k clusters)
- **Divisive:**
    - Comença amb un, tot *inclusive* cluster
    - Cada iteració divideix el cluster fins que cada cluster tingui un punt (o hi hagi k clusters)

Utilitza la **similaritat** o la matriu de distància per fusionar o dividir cada cop

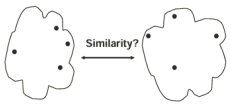
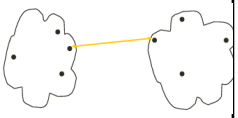
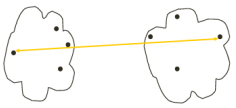
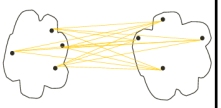
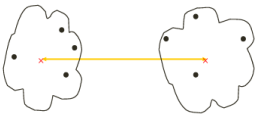
Agglomerative:

63/115
Tècnica més popular.
Operació clau: computació proximitat dels clusters
Algorisme:

1. Compute the proximity matrix
2. Let each data point be a cluster
3. **Repeat**
4. Merge the two closest clusters
5. Update the proximity matrix
6. **Until** only a single cluster remains

Start with clusters of individual points and a proximity matrix	some merging steps, we have some clusters	We want to merge the two closest clusters (C2 and C5) and update the proximity matrix.	The question is "How do we update the proximity matrix?"
			

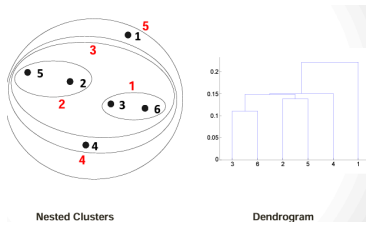
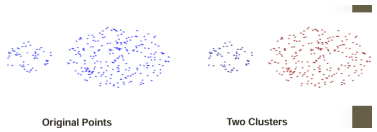
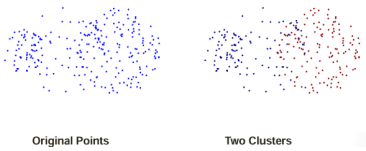
Definir Inter-Cluster:

	min	max	mitjana grup	distància centroids
				

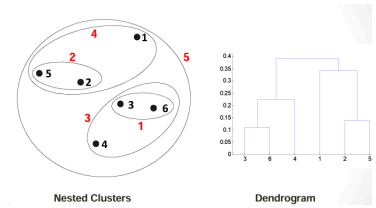
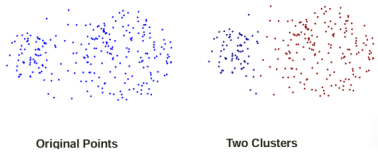
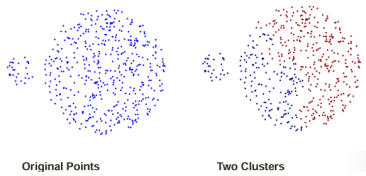
Cost Espai i Temps:

- O(N2) space since it uses the proximity matrix. - N is the number of points.
- O(N3) time in many cases - N steps and at each step the size,
  N2, proximity matrix must be updated and searched
  Complexity can be reduced to O(N2log(N)) time for some approaches

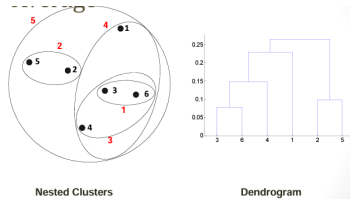
Similitud MIN: entre 2 clusters basat entre els punts més aprop entre els diferents clusters

SINGLE LINK	**fortalesa min**	**limitacions min**
	 *Can handle non-elliptical shapes*	 *Sensitive to noise and outliers*

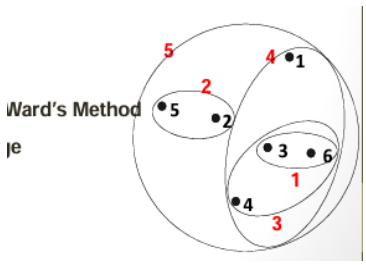
Similitud MAX: entre 2 clusters basat en els dos punts més distants entre els clústers

COMPLETE LINKAGE	**fortalesa max**	**limitacions min**
	 *Less susceptible to noise and outliers*	 *Tends to break large clusters* *Biased towards globular clusters*

Similitud MITJANA Grup: proximitat entre 2 clusters és la mitjana de la parella entre els dos punts dels clusters, utilitzar la mitjana connectivitat per escalabilitat pq la proximitat millora sempre els clústers grans.

COMPROMIS SINGLE** and **COMPLETE LINKAGE	**fortalesa mitjana**	**limitacions mitjana**
	*Less susceptible to noise and outliers*	*Biased towards globular clusters*

Similitud WARDS METHOD: entre 2 clusters basat en l'augment en l'error quadratic quan 2 clusters son fusionats

	fortalesa wards	**limitacions wards**
	*Less susceptible to noise and outliers*	*Biased towards globular clusters*

Problemes i Limitacions:

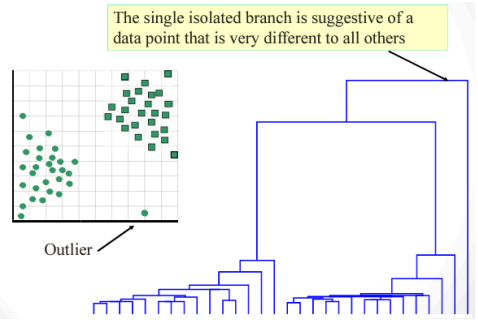
Una vegada feta la decisió combinar 2 clusters, no es pot desfer

Cap funció objectiva es minimitza directament

Els diferents esquemes tenen problemes amb un o més:

- Sensibilitat al soroll i outliers
- Dificultat per manejar grups de diferents mides i formes convexes grans
- Trencar grans clusters

Ajuda a detectar **outliers** →

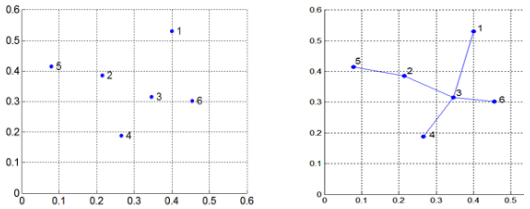


MST: Divisive Hierarchical Clustering:

91/115

Crear un MST - Minimum Spanning Tree

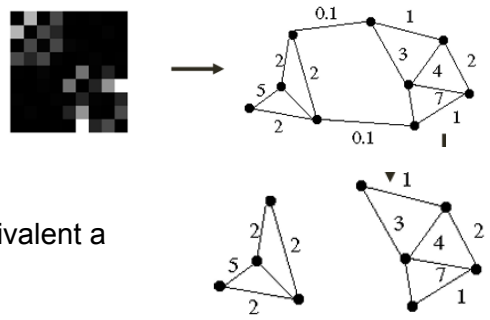
- Començar arbre qualsevol lloc
- En passos successius, troba el parell de punts (p,q) que p sigui concurrent al arbre però q no
- Entrar q al arbre i posar una aresta entre p i q



Spectral/Graph-based Clustering:

Idea:

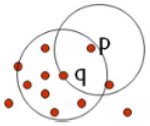
- pensar en la matriu de distàncies com un gràfic amb **pes:**
  - on els objectes són nodes
  - les arestes existeixen per a objectes gran
  - similitud
- trobar una bona agrupació dels objectes és una mica equivalent a trobar subgrafs de baix pes
  - es pot reduir a “tall mínim
  - algorismes relacionats amb l'estructura pròpia de la matriu de distància



DBScan: Algorismes amb Estimadors Densitat:

DBSCAN is a density-based algorithm.

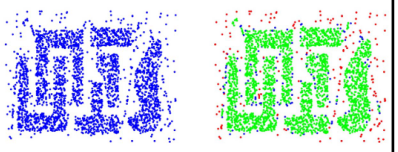
- Density = number of points within a specified radius (Eps)
- A point is a **core point** if it has more than a specified number of points (MinPts) within Eps
  - These are points that are at the interior of a cluster
- A **border point** has fewer than MinPts within Eps, but is in the neighborhood of a core point
- A **noise point** is any point that is not a core point or a border point.



- Eliminate noise points
- Perform clustering on the remaining points

```
current_cluster_label ← 1
for all core points do
  if the core point has no cluster label then
    current_cluster_label ← current_cluster_label + 1
    Label the current core point with cluster label current_cluster_label
  end if
  for all points in the Eps-neighborhood, except i^th the point itself do
    if the point does not have a cluster label then
      Label the point with cluster label current_cluster_label
    end if
  end for
end for
```

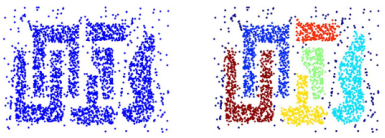
On realitza bons resultats:



Original Points

Point types: core, border and noise

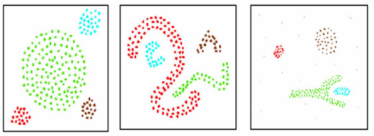
Eps = 10, MinPts = 4



Original Points

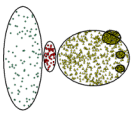
Clusters

- Resistant to Noise and eliminate outliers from clusters.
- Can handle clusters of different shapes and sizes



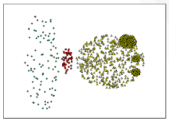
- DBSCAN does not require the number of clusters a priori
- DBSCAN requires just two parameters, and is mostly insensitive to the ordering of the points in the database

On realitza mals resultats:

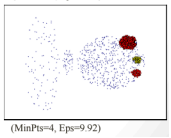


Original Points

- Varying densities
- High-dimensional data



(MinPts=4, Eps=9.75).

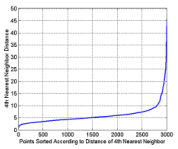


(MinPts=4, Eps=9.92)

Idea is that for points in a cluster, their k^{th} nearest neighbors are at roughly the same distance

Noise points have the kth nearest neighbor at farther distance

So, plot sorted distance of every point to its kth nearest neighbor



<u>Evolució dels clusters:</u>

105/115

Per *supervised classification* mesures variades per **evaluar quan bo és el model:** *Accuracy,Precision Recall*

Per:

- evitar trobar patrons erronis
- comparar clusters algorismes
- comparar 2 sets de clusters
- comparar 2clusters

Mesures numériques aplicades per avaluar x aspectes de la validació:

CRITERIA (estrategia general) o ÍNDEX (mesura numèrica impementa CRITERI)

- **External index:** quines etiquetes del cluster es igual externament a les etiquetes externes donades (Entropia)
- **Internal index:** Mesura que bo és el clúster sense saber informació externa(SSE)
- **Relative index:** comparar 2 clusters diferents (External o internal index)

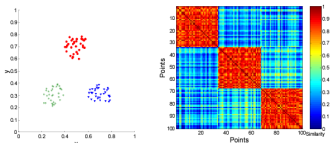
Mesurar cluster Validant via CORRELACIÓ:

Two matrices
- Proximity Matrix
- "Incidence" Matrix
  - One row and one column for each data point
  - An entry is 1 if the associated pair of points belong to the same cluster
  - An entry is 0 if the associated pair of points belongs to different clusters

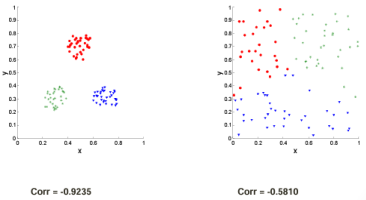
Compute the correlation between the two matrices
- Since the matrices are symmetric, only the correlation between n(n-1) / 2 entries needs to be calculated.

High correlation indicates that points that belong to the same cluster are close to each other.

Not a good measure for some density or contiguity based clusters.



Order the similarity matrix with respect to cluster labels and inspect visually

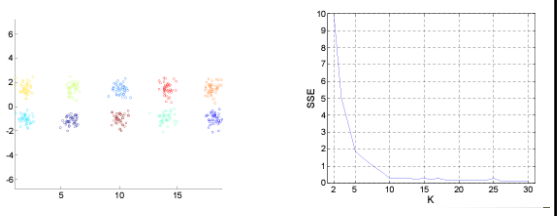


Correlation of incidence and proximity matrices for the Kmeans clusterings of the following two data sets

Internal Measure: SSE

Clusters in more complicated figures aren't well separated ***Internal Index:*** *Used to measure the goodness of a clustering structure without respect to external information*
SSE is good for comparing two clusteringsor two clusters (average SSE).

also used to estimate the number of clusters



Internal Measure: COHESION & SEPARATION

Cluster Cohesion: Measures how closely related are objects in a cluster
Cluster Separation: Measure how distinct or well-separated a cluster is from other clusters

Cohesion is measured by the within cluster sum of squares (SSE)

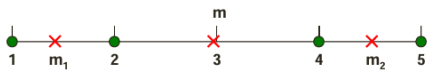
$$WSS = \sum_i \sum_{x \in C_i} (x - m_i)^2$$

Separation is measured by the between cluster sum of squares

$$BSS = \sum |C_i| (m - m_i)^2$$

- Where $|C_i|$ is the size of cluster i

- Example: SSE
  - BSS + WSS = constant



K=1 cluster:

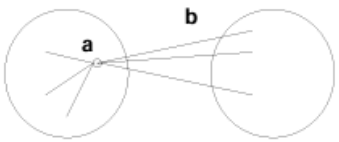
$$WSS = (1-3)^2 + (2-3)^2 + (4-3)^2 + (5-3)^2 = 10$$
$$BSS = 4 \times (3-3)^2 = 0$$
$$Total = 10 + 0 = 10$$

K=2 clusters:

$$WSS = (1-1.5)^2 + (2-1.5)^2 + (4-4.5)^2 + (5-4.5)^2 = 1$$
$$BSS = 2 \times (3-1.5)^2 + 2 \times (4.5-3)^2 = 9$$
$$Total = 1 + 9 = 10$$

Internal Measure: Silhouette Coefficient

Silhouette Coefficient combine ideas of both **cohesion** and **separation**, but for **individual points**, as well as **clusters and clusterings**
For an individual point, i
• Calculate a= average distance of ito the points in cluster
• Calculate b= min (average distance of i to points another cluster)
• The silhouette coefficient for a point is then given by s = 1 –a/b if a < b, (or s = b/a -1 if a b, not the usual case)



Typically between 0 and 1. The closer to 1 the better.

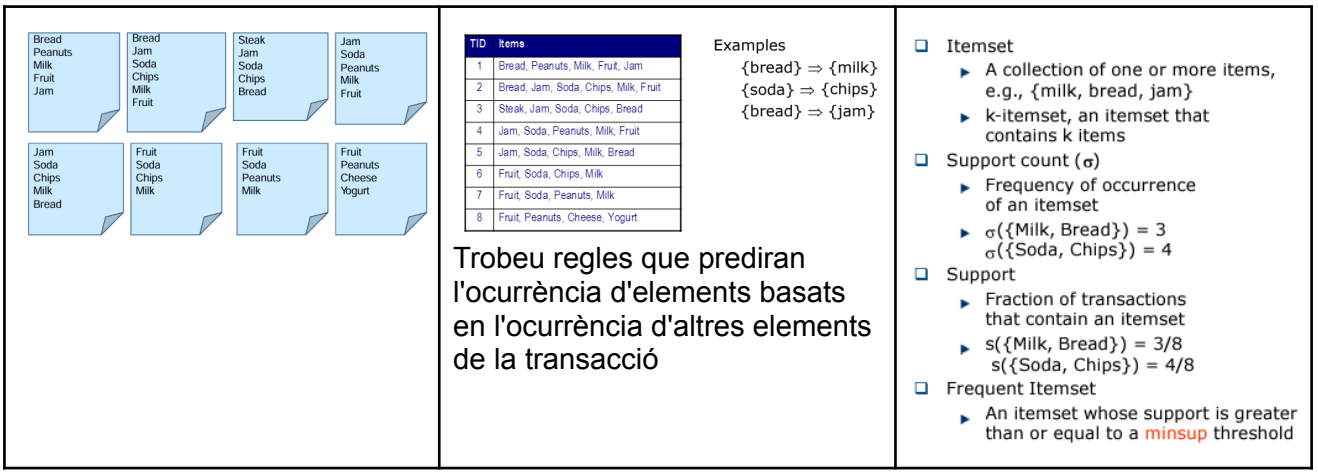
Tema 7: Regles d'Associació

1/38

Association mining: **Trobar patrons** freqüents, associacions, correlacions o estructures causals entre **conjunts d'elements** o objectes en **bases de dades** de transaccions, bases de dades relacionals i altres dipòsits d'informació

Aplicacions:

- Anàlisi de cistella de mercat: donada una base de dades de transaccions de clients, on cada transacció és un conjunt d'articles, l'objectiu és trobar grups d'articles que sovint es compren junts.
- Telecomunicacions: (cada client és una transacció que conté el conjunt de trucades telefòniques)
- Targetes de crèdit/Serveis bancaris: (cada targeta/compte és una transacció que conté el conjunt de pagaments del client)
- Tractaments mèdics: (cada pacient es representa com una transacció que conté el conjunt ordenat de malalties)
- Anàlisi del joc de bàsquet: (cada joc es representa com una transacció que conté el conjunt ordenat de passades de pilota)



Regla d'associació:

IMPLICACIÓ de la forma que X ⇒ Y on X i Y son itemset

Ex: bread⇒milk

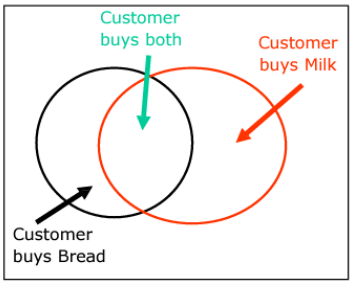
Norma avaluació mètrica, Support i Confiança:

Support (s): Fracció de les transaccionss que contenen tant X i Y

$$s = \frac{\sigma(\{\text{Bread, Milk}\})}{\#\text{ of transactions}} = 0.38$$

Confidence (c): Mesures quan sovint items en Y apareixen en transaccions que contenen X

$$c = \frac{\sigma(\{\text{Bread, Milk}\})}{\sigma(\{\text{Bread}\})} = 0.75$$



Objectiu: donat set de transaccions T, l'objectiu regla d'associació es trobar totes normes q:

- support ≥mindup threshold
- confidence ≥ minconf threshold

Utilitzant força bruta: → Computacionalment prohibit!

- llista totes les possibles regles associació
- Calcular tots els supports i confidence per cada regla
- Elimina les regles que fallin el llindar minsup i minconf

11/38

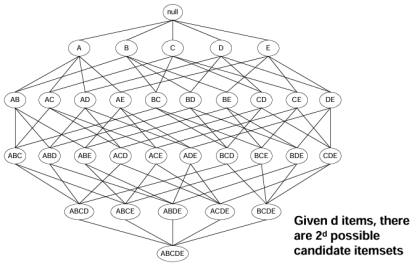
{Bread, Jam} ⇒ {Milk} s=0.4 c=0.75
{Milk, Jam} ⇒ {Bread} s=0.4 c=0.75
{Bread} ⇒ {Milk, Jam} s=0.4 c=0.75
{Jam} ⇒ {Bread, Milk} s=0.4 c=0.6
{Milk} ⇒ {Bread, Jam} s=0.4 c=0.5

All the above rules are binary partitions of the same itemset:

{Milk, Bread, Jam}

2 Step approach:

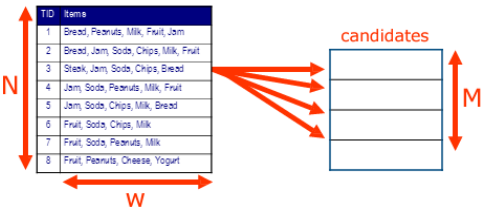
1. itemsets frequent generation: generar tots els items set que support ≥mindup
2. Generació de regles:
   - Generar regles alta confidence del itemsets frequent
   - Cada regla es partició binària al itemsets frequent



itemsets frequent es computacionalment car

Força bruta approach:

- Cada itemset en la matriu candidat per itemset frequent
- Contar el support per cada candidat a través bd
- Igualar cada transacció en contra cada candidat
  Cost O(NMw) -> car si M=2^^d



Estratègies:

- **Reduir nombre de candidats (M)**
  Cerca completada: M=2^^d → *Pruning techniques* reduir M
- **Reduir nombre de transaccions (N)**
  Reduir tamany N aquanb el tamany de itemset creix
- **Reduir nombre de comparacions (NM)**
  Utilitzar estructures de dades efficients per guardar candidats/transaccions
  No cal igualar cada candidat per cada transacció

Reduir nombre de Candidats:

<u>Apriori principle:</u>

18/38

Itemset es frequent, tots els subsets també han de ser frequents

Segueix propietat de suport: $\forall X, Y : (X \subseteq Y) \Rightarrow s(X) \geq s(Y)$

Suport d'un itemset mai pot superar els seus subsets - **anti-monotone property of support**

Algorisme:

- Let k=1
- Generate frequent itemsets of length 1
- Repeat until no new frequent itemsets are identified
  - Generate length (k+1) candidate itemsets from length k frequent itemsets
  - Prune candidate itemsets containing subsets of length k that are infrequent
  - Count the support of each candidate by scanning the DB
  - Eliminate candidates that are infrequent, leaving only those that are frequent

C_k: Candidate itemset of size k

L_k : frequent itemset of size k

```
L1 = {frequent items};
for (k = 1; Lk !=∅; k++) do begin
    Ck+1 = candidates generated from Lk;
   for each transaction t in database do
        increment the count of all candidates in Ck+1
        that are contained in t
   Lk+1  = candidates in Ck+1 with min_support
   end
return ∪k Lk;
```

Join Step

- C_k is generated by joining L_{k-1} with itself

Prune Step

- Any (k-1)-itemset that is not frequent cannot be a subset of a frequent k-itemset

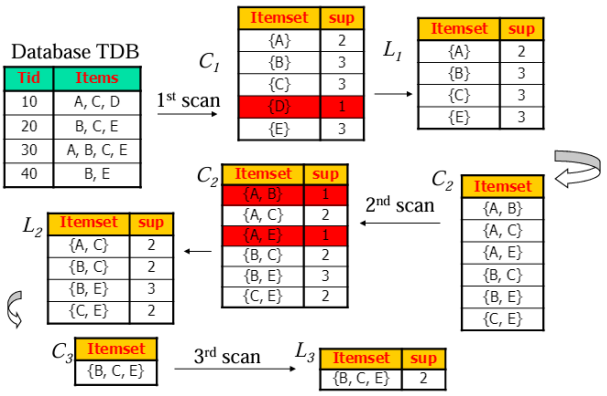
<u>Un candidat i test approach:</u>

- Qualsevol subset d'un itemset frequent ha de ser frequent

(if {beer, diaper, nuts}is frequent, so is {beer, diaper} every transaction having {beer, diaper, nuts} also contains {beer, diaper})

- Apriori pruning princple: Si hi ha **algun itemset** que es **infrequent** el superset no hauria de generar-se/provar
- Mètode:
  - Generar llargària k+1 candidats itemset de llargaria k items frequents
  - test candidats en contra database db

Algorisme eficient i escalable



<u>Generar normes:</u>

Donat un itemset frequent L, trobar tots els subsets no buits tal que f→ L - f satisfà minimun confidence

If {A,B,C,D} is a frequent itemset, candidate rules:

ABC →D, ABD → C, ACD→B, BCD →A, A →BCD, B →ACD, C →ABD, D →ABC, AB →CD, AC → BD, AD → BC, BC →AD, BD →AC, CD →AB

If |L| = k, then there are $2^k - 2$ candidate association rules (ignoring L → ∅ and ∅ → L)

Crear minsup:

Si molt alt, prendre itemsets que tinguin items interessants/raros

Si molt baix, computacionalment car i nombre itemsets és molt gran

Un sol minsup pot no ser efectiu

Exemple:

Let's assume we have a Database about behavior of 5.000 students and that from the dataset, we know:

- 3.000 of them play basketball (P) [60%]
- 3.750 eat cereals for breakfast (E) [75%]
- 2.000 play basketball and eat cereals ($P \wedge E$) [40%]

Let's set thresholds: s=40%, c=60%

Can we obtain the following rule? $P \rightarrow E$

$S(P \wedge E) \geq 40\%$

$S(P \wedge E)/S(P) = 0{,}4/0{,}6 = 66\% > 60\%$

So rule says: If the student plays basketball, then with confidence of 0.66 will eat cereals.

But notice that we know S(E) is 75%... So rule does not contribute to our knowledge

We should filter out trivial and misleading associations $A \rightarrow B$:

$$\boxed{\frac{S(A \wedge B)}{S(A)} - S(B) > 0}$$

In the Basketball example:

$S(P \wedge E)/S(P) - S(E) = 0{,}66 - 0{,}75 = -0{,}09 < 0$

-No te res a veure q juguin a basquet i mengin cereals