# From Data Mining to Big Data & Data Science: a Computational Perspective

**Miquel Sànchez-Marrè**

Knowledge Engineering & Machine Learning Group (KEMLG)

Dept. de Llenguatges i Sistemes Informàtics

Universitat Politècnica de Catalunya-Barcelona**Tech**

**miquel@lsi.upc.edu**

**http://www.lsi.upc.edu/~miquel**

19 de Julio 2013

**https://kemlg.upc.edu**

Knowledge Engineering and Machine Learning Group
UNIVERSITAT POLITÈCNICA DE CATALUNYA

# Content

- Introduction
- Knowledge Discovery in Databases
  - Data Pre-processing
  - Data Mining Techniques: Statistical & ML methods
  - Data Post-Processing
- Big Data / Data Science
- Social Mining
- Scaling from Data Mining to Big Mining
  - Computational Techniques
  - Examples of Extrapolation of classical DM Techniques
- A look to Mahout
- Big Data Tools & Resources
- Big Data Trends

# INTRODUCTION

https://kemlg.upc.edu

Knowledge Engineering and Machine Learning Group
UNIVERSITAT POLITÈCNICA DE CATALUNYA

# Complex Real-world Systems/Domains

- Exist in the daily real life of human beings, and normally show a *strong complexity* for their understanding, analysis, management or solving.
- They imply several **decision making tasks** very *complex* and *difficult*, which usually are faced up by human experts
- Some of them, in addition, could have ***catastrophic consequences*** either for human beings or for the environment or for the economy of one organization
- Examples
  - Environmental System/Domains
  - Medical System/Domains
  - Industrial Process Management Systems/Domains
  - Business Administration & Management Systems/Domains
    - Marketing
    - Decisions on products and prices
    - Decisions on human resources
    - Decisions on strategies and company policy
  - ***Internet***

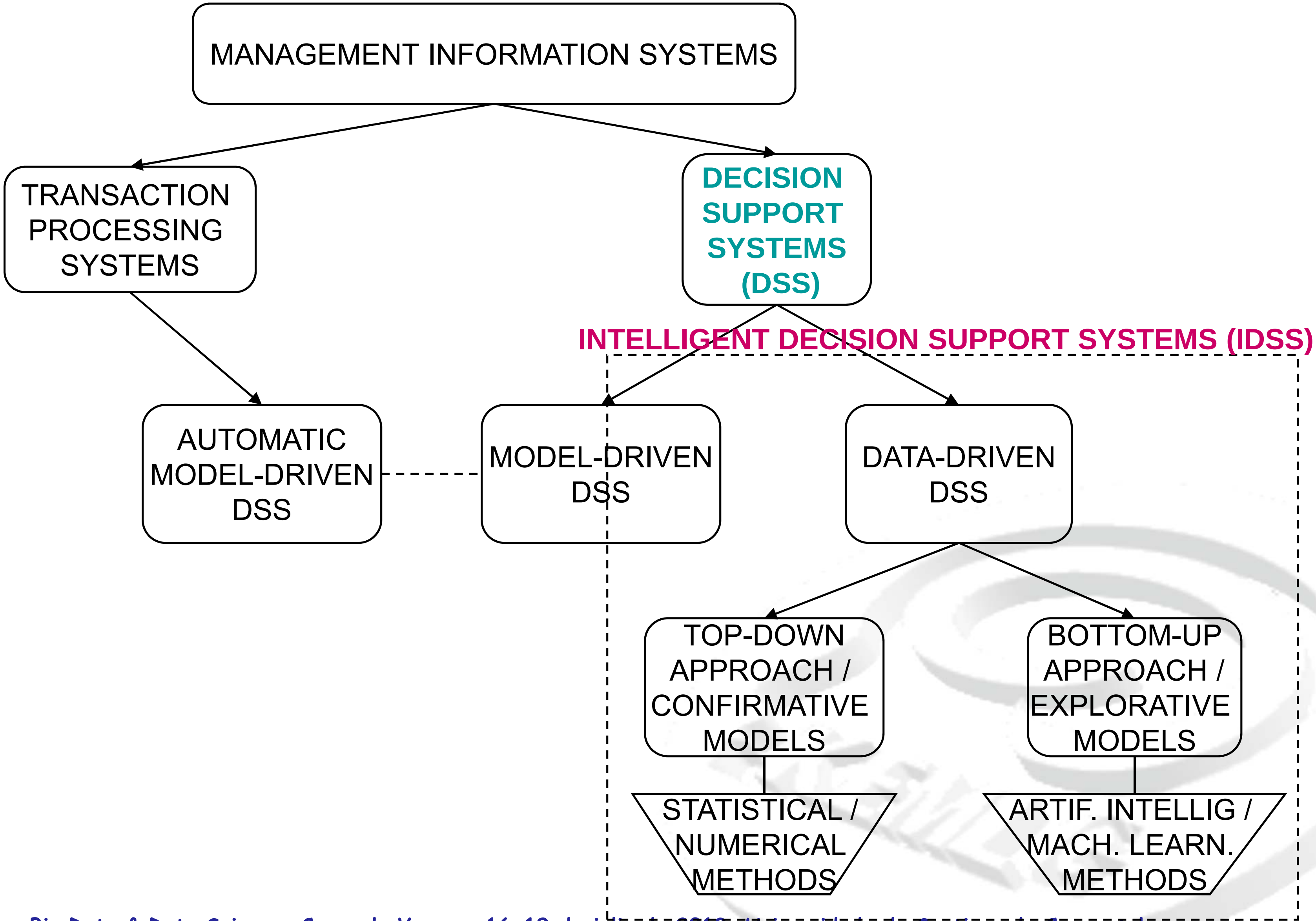
# Need for Decision Making Support Tools

- Complexity of the decision making process
- Accurate Evaluation of multiple alternatives
- Need for forecasting capabilities
- Uncertainty
- Data Analysis and Exploitation
- Need for including experience and expertise (knowledge)

⬇

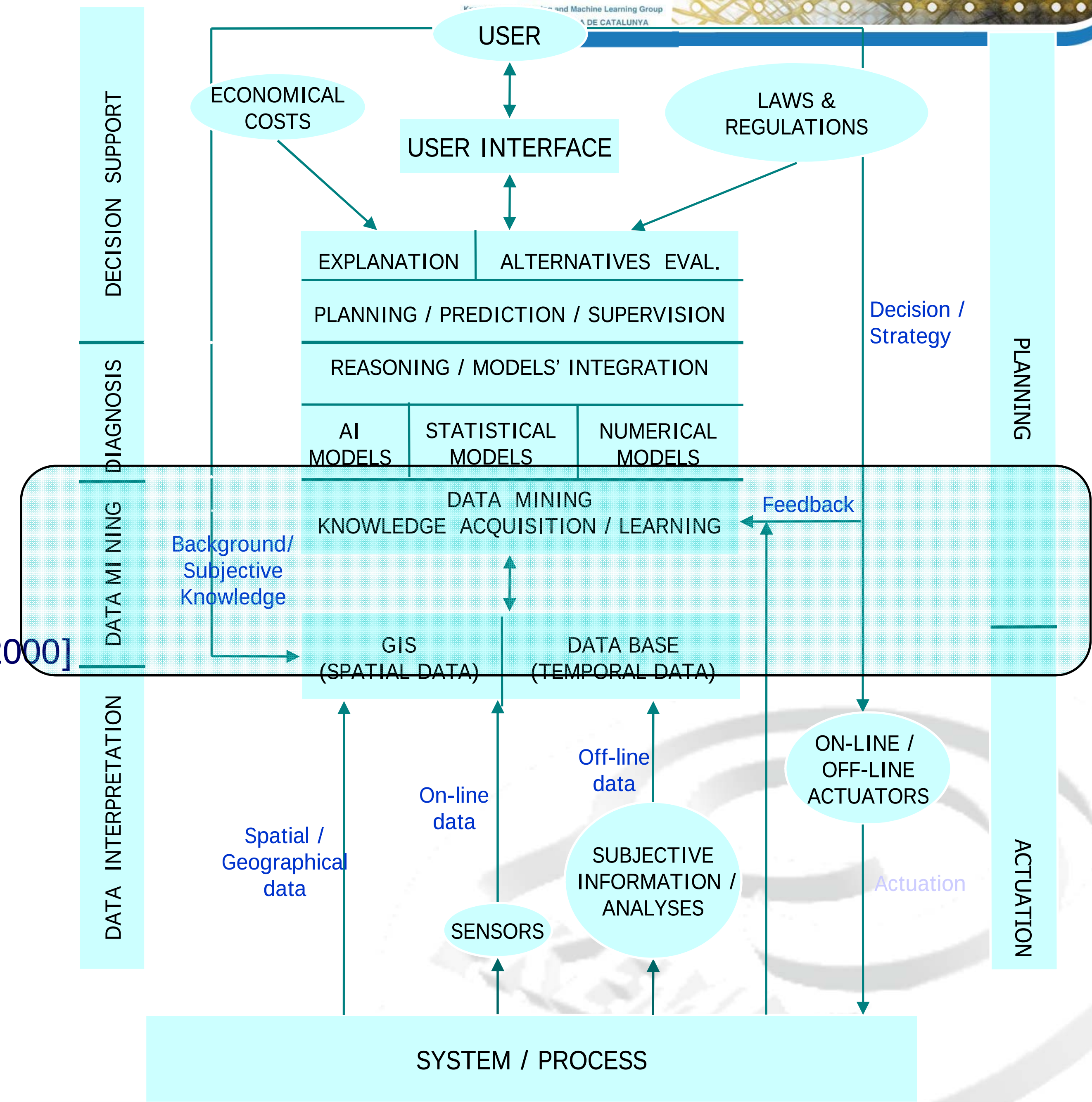
- Computational tools: Decision Support Systems (DSS)
- Intelligent Computational Tools: Intelligent Decision Support Systems (IDSS)

# Management Information Systems



# Intelligent Decision Support Systems (IDSS)

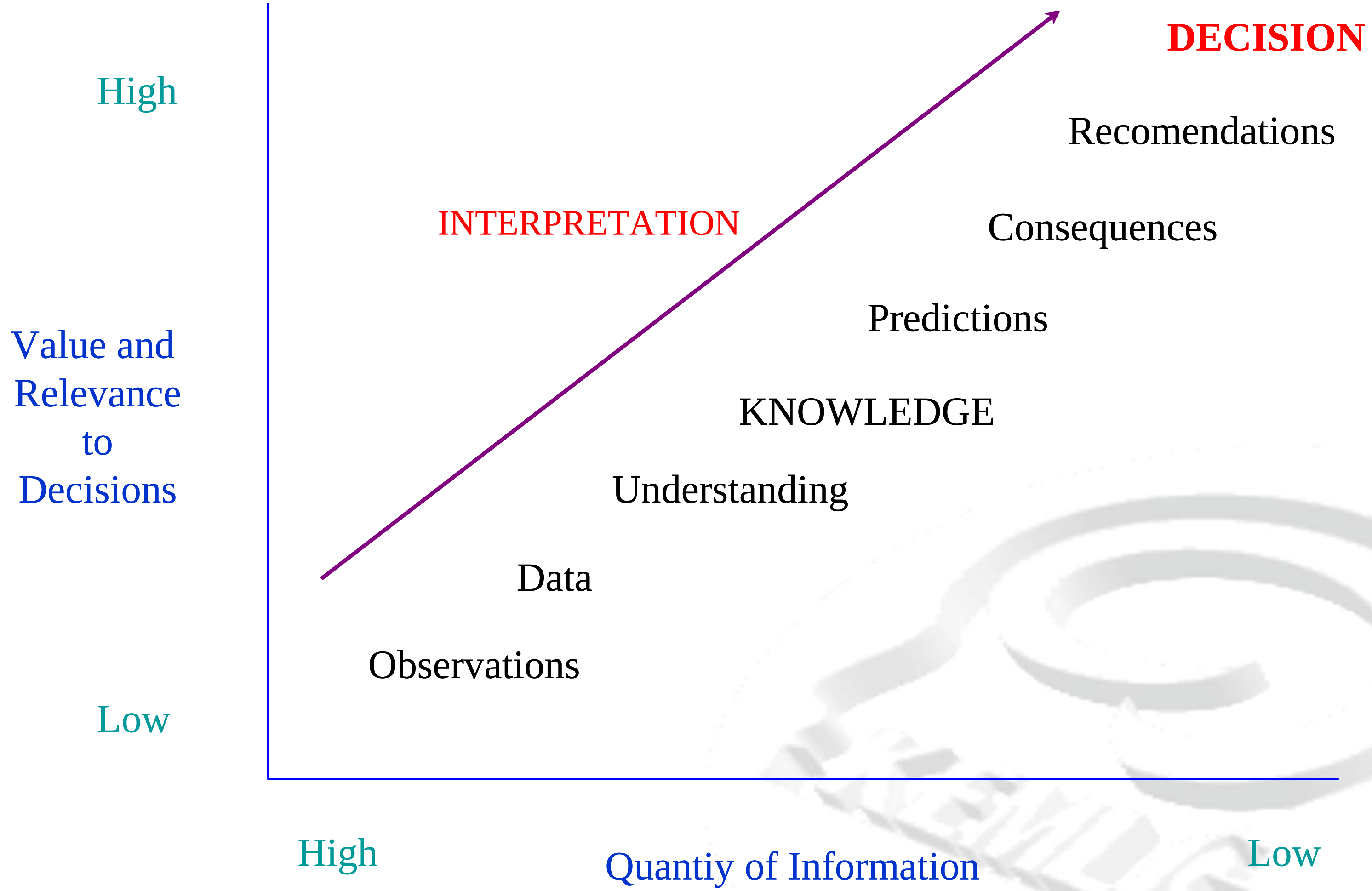
[Sànchez-Marrè et al., 2000]



# Why Data Mining is important ?

## From observations to decisions

[Adapted from A.D. Witakker, 1993]



# KNOWLEDGE DISCOVERY / DATA MINING

## Intelligent Data Analysis

https://kemlg.upc.edu

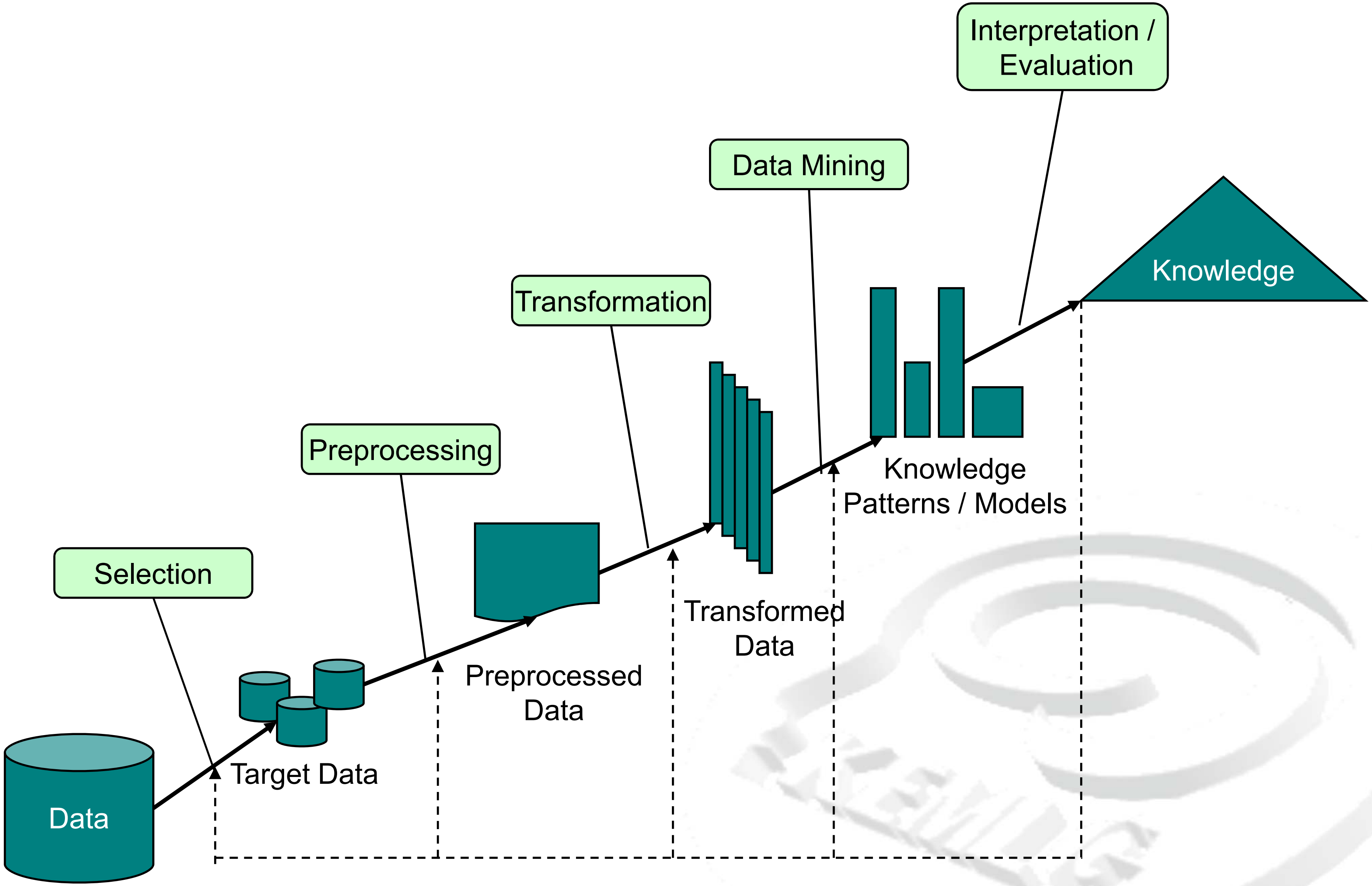
Knowledge Engineering and Machine Learning Group
UNIVERSITAT POLITÈCNICA DE CATALUNYA

# KDD Concepts

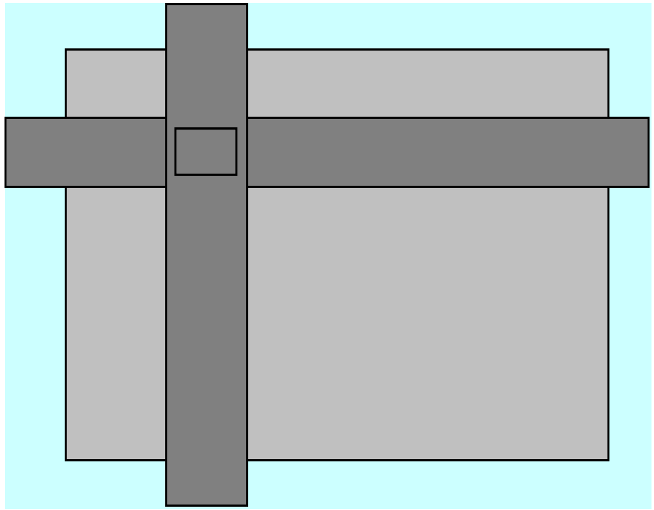
- KDD
  - Knowledge Discovery in Databases is the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data [Fayyad et al., 1996]
  - KDD is a multi-step process
- Data Mining
  - Data Mining is a step in the KDD process consisting of particular data mining algorithms that, under some aceptable computational efficiency limitations, produces a particular set of patterns over a set of examples or cases or data. [Fayyad et al., 1996]

# The KDD process

[Fayyad, Piatetsky-Shapiro & Smyth, 1996]



# Terminology



| Data Bases | Artificial Intelligence | Statistics |
|---|---|---|
| Table | Data Matrix | Data |
| Register | Instance/Example | Observation/Individual |
| Field | Attribute/Feature | Variable |
| Value | Data | Value |

# DATA PRE-PROCESSING

# Problems with data

- Huge amount of data
  - Corrupted Data
  - Noisy Data
  - Irrelevant Data / Attribute Relevance
  - Attribute Extraction
  - Numeric and Symbolic Data
- Scarce Data
  - Missing Attributes
  - Missing Values
- Fractioned Data
  - Incompatible Data
  - Different Source Data
  - Different Granularity Data

# Data Preparing Data

- Data Tranformation
  - Data Filtering
  - Data Sorting
  - Data Edition
  - Noise Modelling
- New Information Obtention
  - Visualization
  - Removing
  - Data Selection
  - Sampling
- New Information Generation
  - Data Engineering
  - Data Fusion
  - Time Series Analysis
  - Constructive Induction

# Data Cleaning

- Duplicated data removing
- Data Inconsistency solving
- Outlier Detection and Management
- Error Detection and Management

# Coding

- Codify some fields and generate a code table
  - Address to region
  - Phone prefix to province code
  - Birth data to age, and age to decade
  - To normalize huge magnitudes: millions to thousands (account balance, credits, ...)
  - Sex, married/single, … transformed to binary
  - Weekly, monthly informations, … to number of week, number of month, …

# Flattering

- Transform an attribute of cardinality *n* in *n* binary attributes to get a unique register per element

| DNI | hobby |
|---|---|
| 45464748 | sport |
| 45464748 | music |
| 45464748 | reading |
| 50000001 | music |

→

| DNI | Spo | Mus | Read |
|---|---|---|---|
| 45464748 | 1 | 1 | 1 |
| 50000001 | 0 | 1 | 0 |

# Data Selection

- Mechanism to reduce the size of the data matrix, by means of the following techniques:
  - Instance Selection
  - Feature Selection
  - Feature Weighting (Feature Relevance Determination)

# Instance Selection

- Goal: to reduce the number of examples
- Methods [Riaño, 1997]:
  - Exhaustive methods
  - Greedy methods
  - Heuristic methods
  - Convergent methods
  - Probabilistic methods

# Feature Selection

- Goal: to reduce the number of features (dimensionality)
- Categories of methods:
  - Feature ranking
    - Feature ranking methods ***ranks the features by a metric*** and eliminates all features that do not achieve an adequate score
  - Subset selection
    - Subset selection methods search ***the set of possible features for the optimal subset.***
    - Methods
      - *Wrapper methods (Wrappers)*: utilize the ML model of interest as a black box to score subsets of feature according to their predictive power.
      - *Filter methods (Filters):* select subsets of features as a preprocessing step, independently of the chosen predictor.
      - *Embedded methods*: perform feature selection in the process of training and are usually specific to given ML techniques

# Feature Weighting

- Goal: to determine the weight or relevance of all features in an automatic way
- Framework for Feature Weighting methods [Wettschereck et al., 1997]:

| **Dimension** | **Possible Value** |
|---|---|
| Bias | {Feedback, Preset} |
| Weights Space | {Continuous, Binary} |
| Representation | {Given, Transformed} |
| Generality | {Global, Local} |
| Knowledge | {Poor, Intensive} |

# Feature Weighting / Attribute Relevance (1)

- Supervised methods
  - Filter methods
    - Global methods
      - Mutual Information algorithm
      - Cross-Category Feature Importance
      - Projection of Attributes
      - Information Gain measure
      - Class Value method
    - Local methods
      - Value-Difference Metric
      - Per Category Feature Importance
      - Class Distribution Weighting algorithm
      - Flexible Weighting
      - Entropy-Based Local Weighting

# Feature Weighting / Attribute Relevance (2)

- Wrapper methods
  - RELIEF
  - Contextual Information
  - Introspective Learning
  - Diet Algorithm
  - Genetic algorithms
- Unsupervised methods
  - Gradient Descent
  - Entropy-based Feature Ranking
  - UEB-1
  - UEB-2

# DATA MINING TECHNIQUES



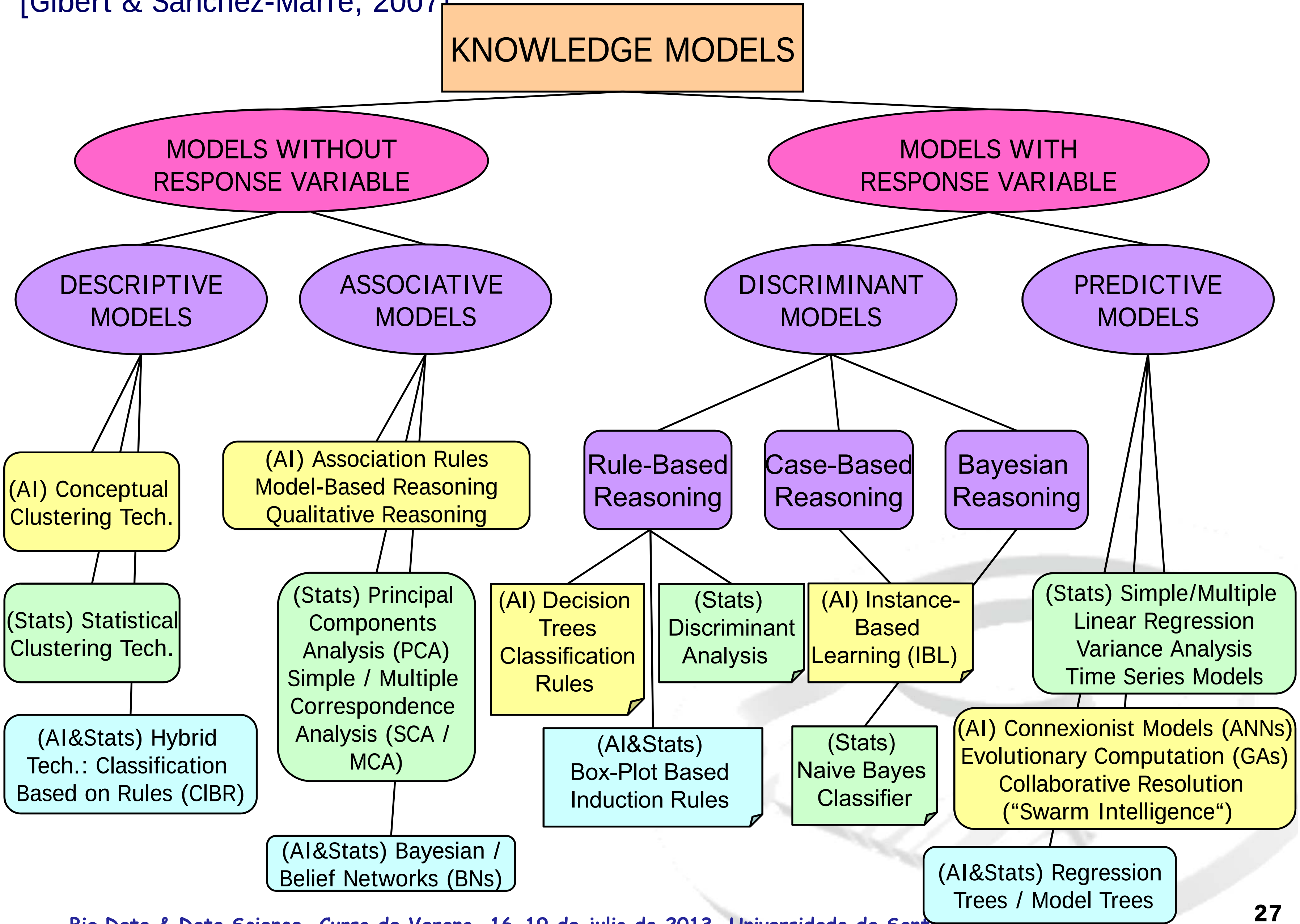
Knowledge Engineering and Machine Learning Group
UNIVERSITAT POLITÈCNICA DE CATALUNYA

# Data Mining Techniques

- Statistical Techniques
  - Linear Models: simple regression, multiple regression
  - Time Series Models (AR, MA, ARIMA)
  - Component Principal Analysis (CPA) / Discriminant Analysis (DA)
- Artificial Intelligence Techniques
  - Decision Trees
  - Classification Rules
  - Association Rules
  - Clustering
  - Instance-Based Learning (IBL, CBR)
  - Connectionist Approach (Artificial Neural Networks)
  - Evolutionary Computation (Genetic Algorithms, Genetic Programming)
- AI&Stats Techniques
  - Regression Trees
  - Model Trees
  - Probabilistic/Belief/Bayesian Networks

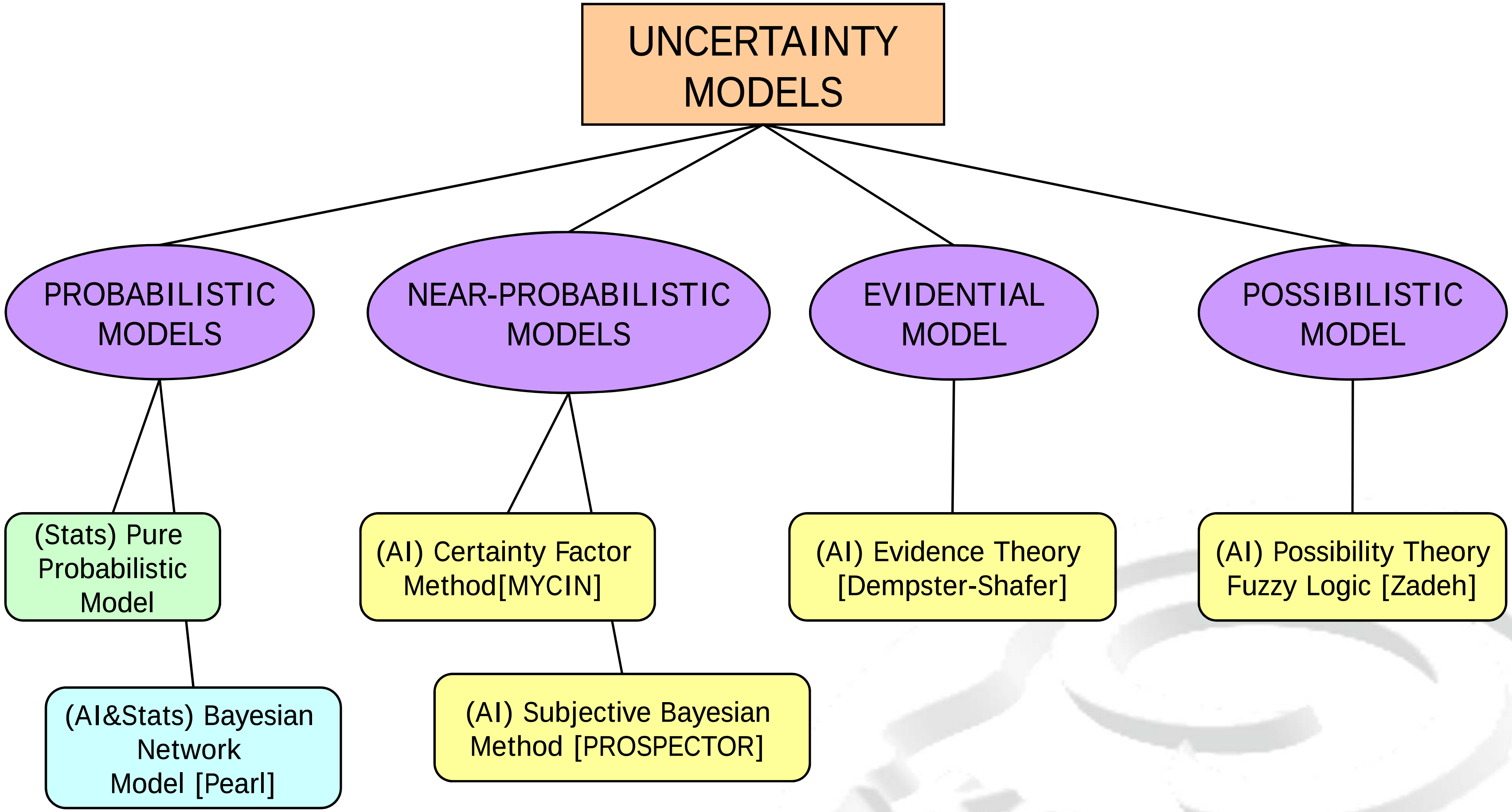
# Model Clasification (1)

[Gibert & Sànchez-Marrè, 2007]



# Model Classification (2)

[Gibert & Sànchez-Marrè, 2007]



# Descriptive Models

## Clustering Techniques

https://kemlg.upc.edu

Knowledge Engineering and Machine Learning Group
UNIVERSITAT POLITÈCNICA DE CATALUNYA

# Clustering / Conceptual Clustering

| | A | B | C |
|---|---|---|---|
| 1 | 0.1 | 0.8 | 0.3 |
| 2 | 0.1 | 0.3 | 100 |
| 3 | 0.7 | 0.3 | 0.45 |
| 4 | 0.3 | 0.38 | 0.42 |

described by A=0.1

described by B $\in$ [0.3,0.38] & C $\in$ [0.42,0.45]

We group {1,2} i {3,4} even though *d(1,2)>d(1,3)*

# Clustering Techniques

- Clustering with K determined clusters
  - K-means method
- Clustering through the neighbours
  - Nearest-Neighbour method (NN method)
- Hierarchical Clustering
- Probabilistic/Fuzzy Clustering
- Clustering based on rules

# K-means Method

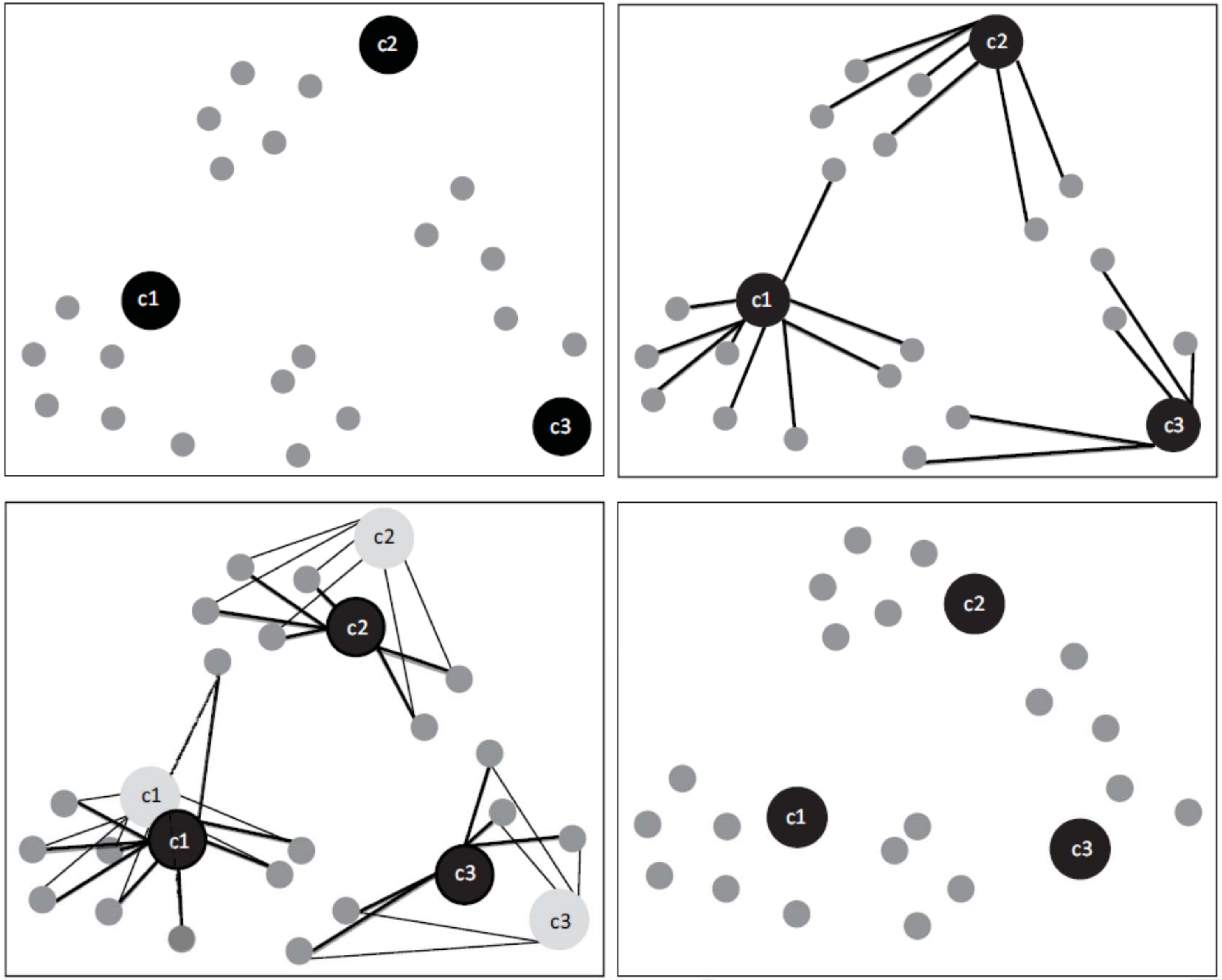
```
Input:  X = {x1, ..., xn}  // Data to be clustered
        k  // Number of clusters

Output:  C= {c1, ..., ck}      // Cluster centroids
```

---

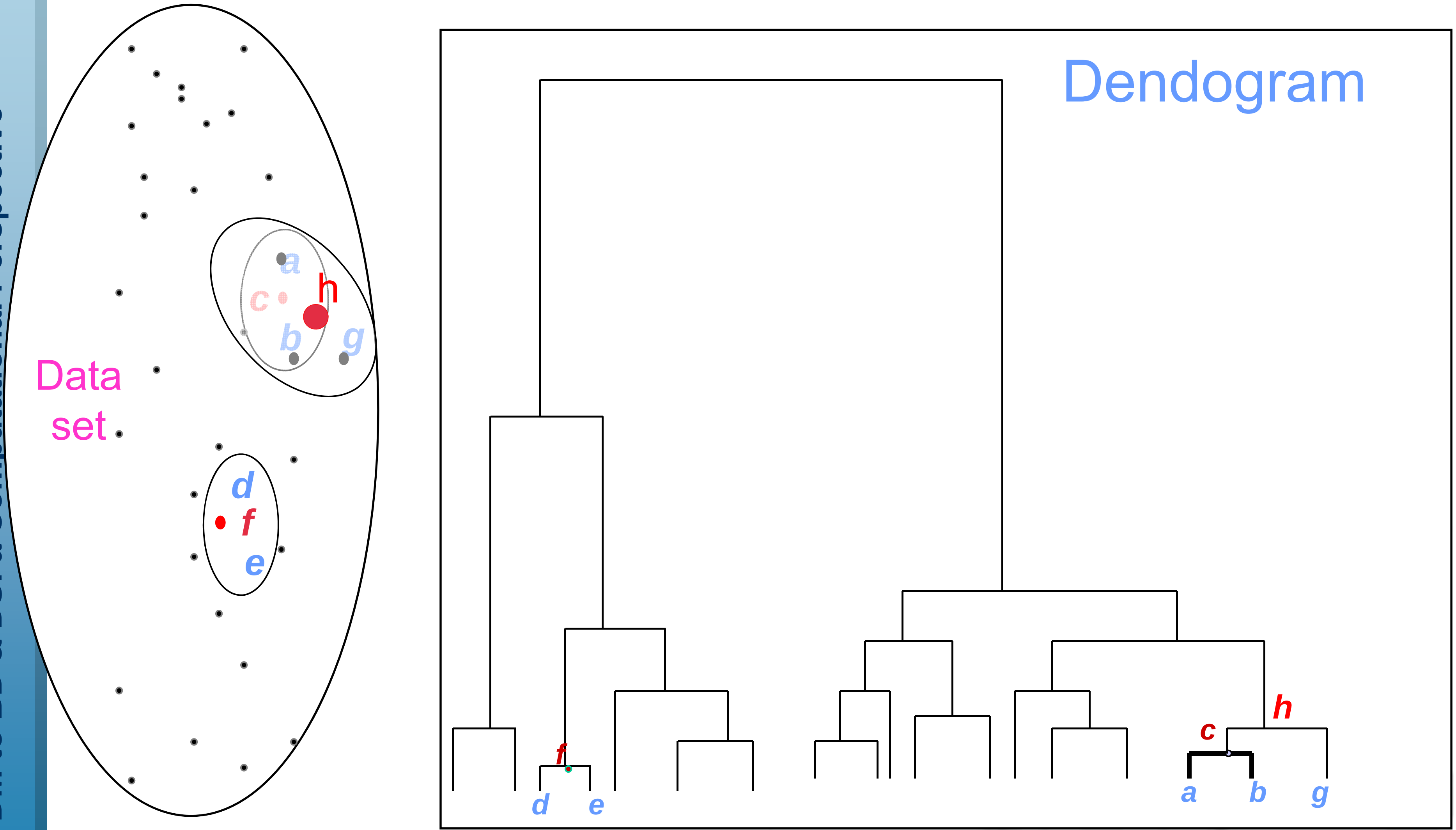
```
Function K-means

initialize C   // random selection from X
while C has changed
   For each xi in X
      cl(xi) = argminj distance (xi, cj)
   endfor
   For each Cj in C
      cj = centroid ({xi | cl(xi) = j})
   endfor
endwhile
return c

End function
```

# K-means method

[Charts extracted from “Mahout in action”, Owen et al., 2011]



# Ascendant hierarchical clustering



# Discriminant Models

## K-NN & Decision trees



https://kemlg.upc.edu

Knowledge Engineering and Machine Learning Group
UNIVERSITAT POLITÈCNICA DE CATALUNYA

# K-NN algorithm

```
Input: T = {t1, ..., tn}  // Training Data points available
       D = {d1, ..., dm}  // Data points to be classified
       k                  // Number of neighbours
Output: neighbours     // the k nearest neigbours
```

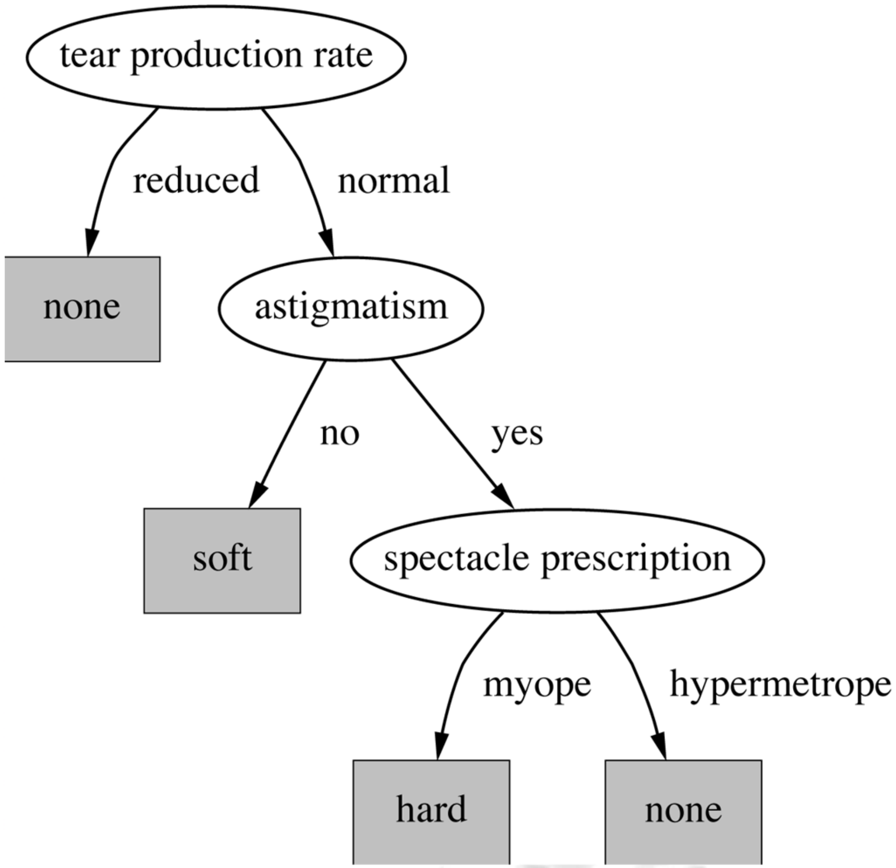
---

```
Function K-NN

Foreach data point d
   neighbours = ∅
   Foreach training data point t
      dist = distance (d,t)
      If |neighbours| < k then
           insert(t, neighbours)
        else
               fartn = argmaxi distance(t,neighboursi)
               if distance (dist < fartn)
                Insert (t, neighbours)
                 Remove (fartn, neighbours)
               endif
`     endif
   endfor
   return majority-vote of K-Nearest (neighbours)
endfor

End function
```

# Decision trees

- The nodes are qualitative attributes
- The branches are the possible values of a qualitative attribute
- The leaves of the tree have the qualitative prediction of the attribute that acts as a class label
- Model the process of deciding to which class belongs a new example of the domain

# Decision Tree: example




Decision tree for the contact lens data

# ID3 algorithm

- **ID3** ≡ Induction Decision Tree [Quinlan, 1979], [Quinlan, 1986]
- Machine Learning Technique
- Decision Tree Induction
- Top-Down strategy
- From a set of **examples/instances** and the class to which they belong, it builds up the *best* decision tree which explains the instances

**Criteria**

- **Compactness** → the more compact better complexity
- **Goodness** → predictive/discriminant ability

# ID3: basic idea

- Select at each step the attribute which can discriminate more.
- The selection is done through maximizing a certain function G (X, A).

# ID3: selection criteria

- Seleccit $A_k$ which *maximizes* the gain of information

$$G(X, A_k) = I(X,C) - E(X, A_k) \Leftrightarrow \mathbf{E(X, A_k) \approx 0}$$

where

information

$$\mathbf{I(X,C)} = -\sum_{C_i \in C} p(X,c_i) * \log_2 p(X,c_i)$$

$$\mathbf{p(X,c_i)} = \# C_i \;/\; \# X$$

Probability that one example belongs to class $C_i$

entropy

$$\mathbf{E(X, A_k)} = \sum_{v_l \in V(A_k)} p(X,v_l) * I(A^{-1}{}_k(v_l),C)$$

$$\mathbf{p(X,v_l)} = \# A^{-1}{}_k(v_l) \;/\; \# X$$

Probability that one example has the value $v_l$ for the attribute $A_k$

# ID3: algorithm

```
Function ID3 (in X,A are sets) returns decision tree is
  var  tree1, tree2 are decision tree endvar
  option
    case  (∃Ci: ∀xj ∈ X --> xj ∈ Ci)  do
       tree1 ← buildTree (Ci)
    case  no (∃Ci: ∀xj ∈ X --> xj ∈ Ci)  do
     option
       case  A ≠ ∅  do
          Amax ← max Ak∈A {G(X,Ak)};
          tree1 ← buildTree(Amax);
          for each  v ∈ V(Amax)  do
            tree2 ← ID3(A-1max (v), A-{Amax});
            tree1 ← addBranch(arbre1, arbre2, v)
          endforeach
       case  A = ∅  do
         tree1 ← buildTree(majorityClass(X))
     endoption
  endoption
  returns tree1
endfunction
```

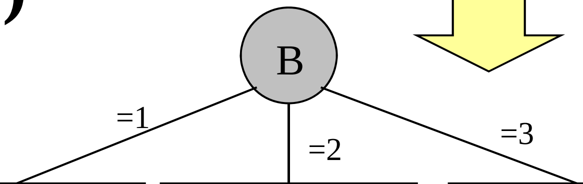
**(1)**

| | A | B | C | D |
|---|---|---|---|---|
| 1. | 1 | 1 | 5 | - |
| 2. | 2 | 1 | 5 | - |
| 3. | 2 | 1 | 7 | + |
| 4. | 1 | 1 | 7 | + |
| 5. | 2 | 2 | 5 | - |
| 6. | 2 | 2 | 7 | - |
| 7. | 1 | 2 | 7 | - |
| 8. | 2 | 3 | 7 | + |

**(2)**

| | p | n | h(p,n) | e |
|---|---|---|---|---|
| A=1 | 1 | 2 | 0,6365 | 0,6593 |
| A=2 | 2 | 3 | 0,6730 | |
| B=1 | 2 | 2 | 0,6931 | **0,3465** |
| B=2 | 0 | 3 | 0,0000 | |
| B=3 | 1 | 0 | 0,0000 | |
| C=5 | 0 | 3 | 0,0000 | 0,4206 |
| C=7 | 3 | 2 | 0,6730 | |

**(3)**




| | A | C | D |
|---|---|---|---|
| 1. | 1 | 5 | - |
| 2. | 2 | 5 | - |
| 3. | 2 | 7 | + |
| 4. | 1 | 7 | + |

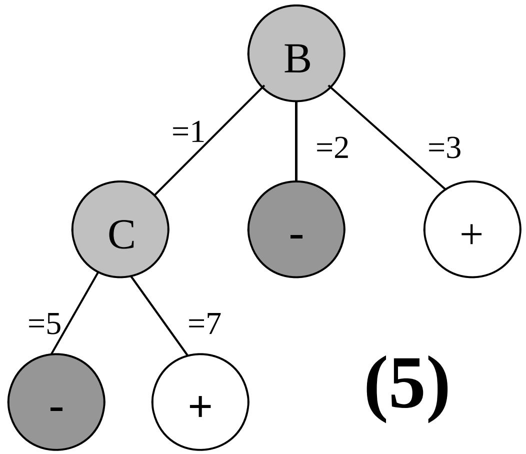
| | | | |
|---|---|---|---|
| 5. | 2 | 5 | - |
| 6. | 2 | 7 | - |
| 7. | 1 | 7 | - |

| | | | |
|---|---|---|---|
| 8. | 2 | 7 | + |

**(4)**

| | p | n | h(p,n) | e |
|---|---|---|---|---|
| A=1 | 1 | 1 | 0,6931 | 0,6931 |
| A=2 | 1 | 1 | 0,6931 | |
| C=5 | 0 | 2 | 0,0000 | **0,0000** |
| C=7 | 2 | 0 | 0,0000 | |

**(5)**



# Data Post-Processing

https://kemlg.upc.edu

Knowledge Engineering and Machine Learning Group
UNIVERSITAT POLITÈCNICA DE CATALUNYA

# Post-processing Techniques (1)

- Post-processing techniques are devoted to transform the direct results of the data mining step into directly understandable, useful knowledge for later decision-making
- Techniques could be summarized in the following tasks [Bruha and Famili, 2000]:
  - ***Knowledge filtering***: The knowledge induced by data-driven models should be normally filtered.
  - ***Knowledge consistency checking***. Also, we may check the new knowledge for potential conflicts with previously induced knowledge.
  - ***Interpretation and explanation***. The mined knowledge model could be directly used for prediction, but it would be very adequate to document, interpret and provide explanations for the knowledge discovered.

# Post-processing Techniques (2)

- ***Visualization***. Visualization of the knowledge (Cox et al., 1997) is a very useful technique to have a deeper understanding of the new discovered knowledge.

- ***Knowledge integration***. The traditional decision-making systems have been dependant on a single technique, strategy or model. New sophisticated decision-supporting systems combine or refine results obtained from several models, produced usually by different methods. This process increases accuracy and the likelihood of success.

- ***Evaluation***. After a learning system induces concept hypotheses (models) from the training set, their evaluation (or testing) should take place.

# Interpretation / Result Evaluation

- Tables summarising data
- Spatial Representation of Data
- Graphical Visualization of Models/Patterns of Knowledge
  - Decision Trees
  - Discovered Clusters
  - Induced Rules
  - Bayesian Network Learned

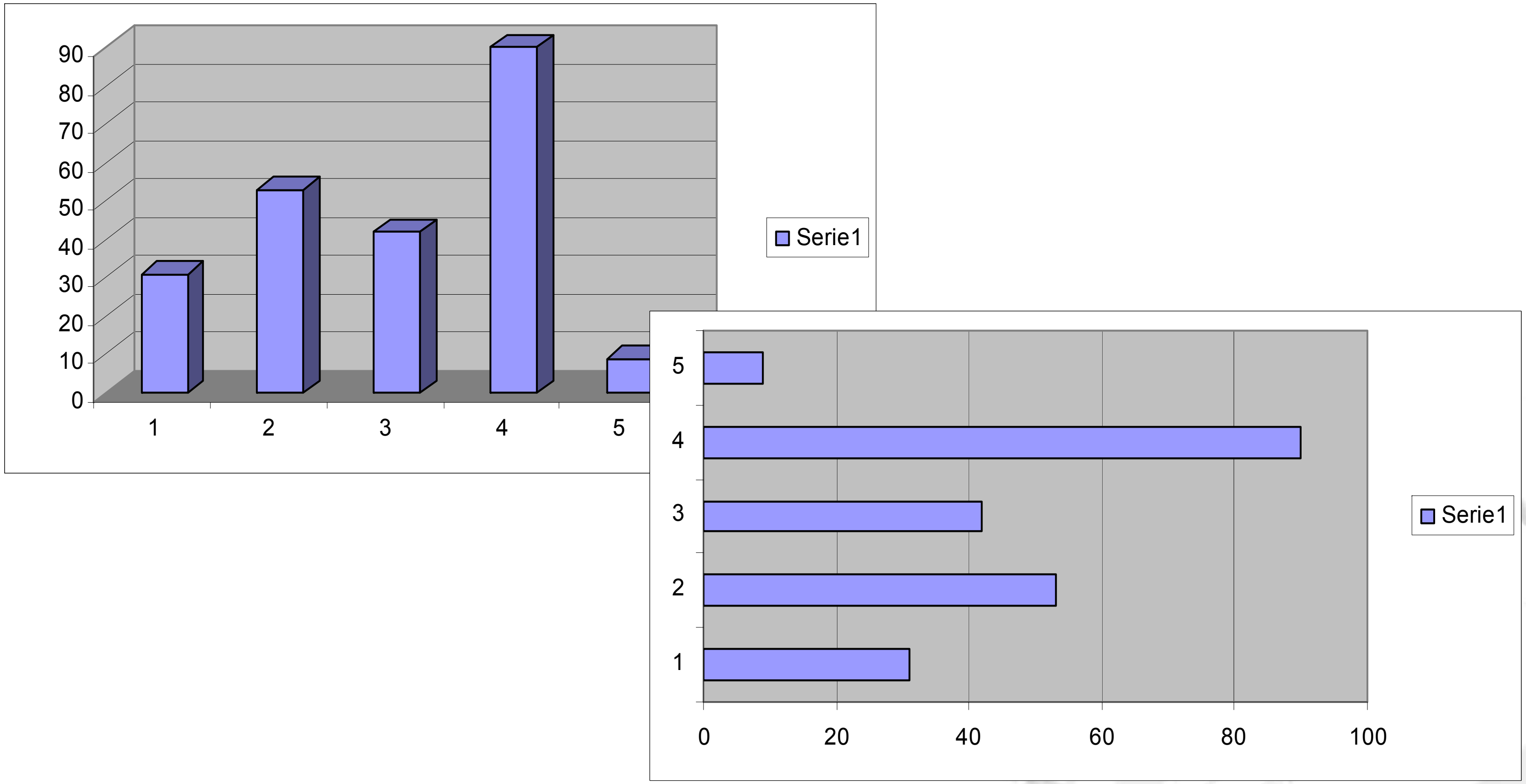
# Result Representation (1)

- Direct Data through tables

| | | |
|---|---|---|
| 0.343689 | 10000 | 5879.9875 |
| 0.467910 | 2345 | 98724.935 |
| 0.873493 | 34 | 92235.9620 |

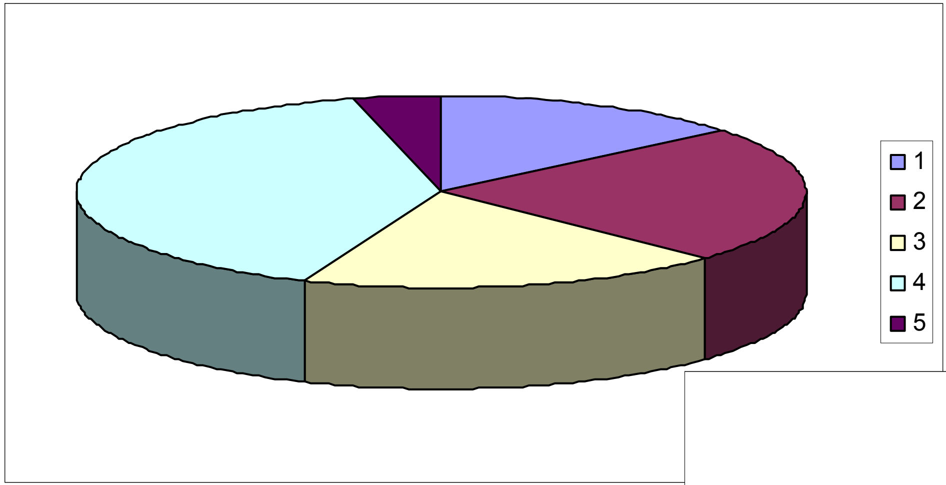
# Result Representation (2)

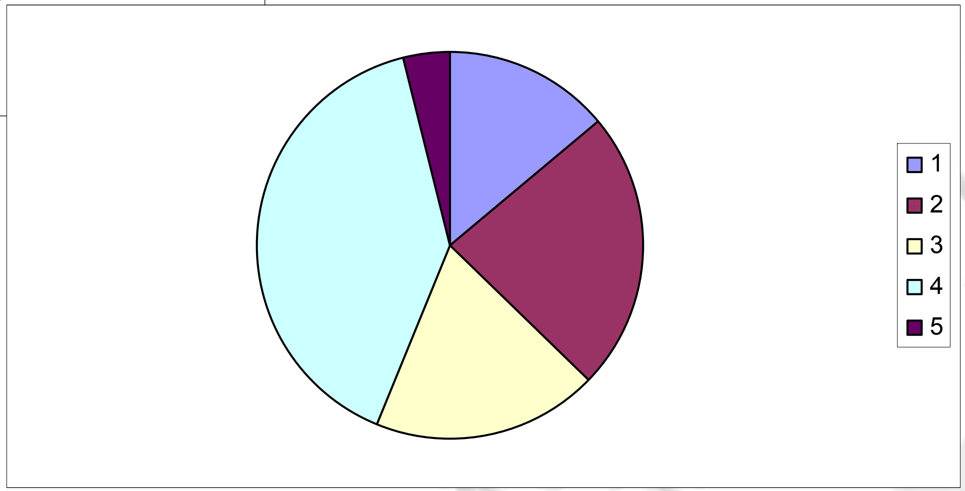
- Features through histograms



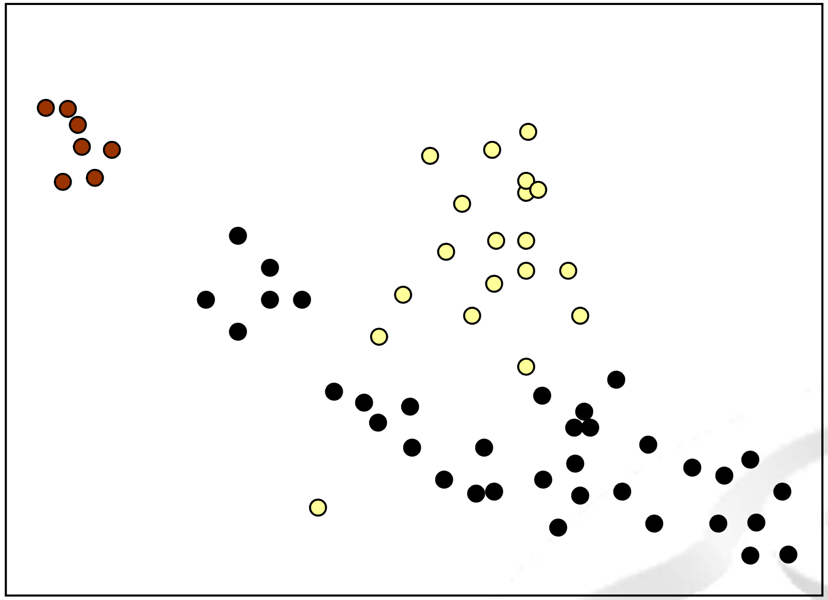
# Result Representation (3)

- Proportions through pie chart graphics

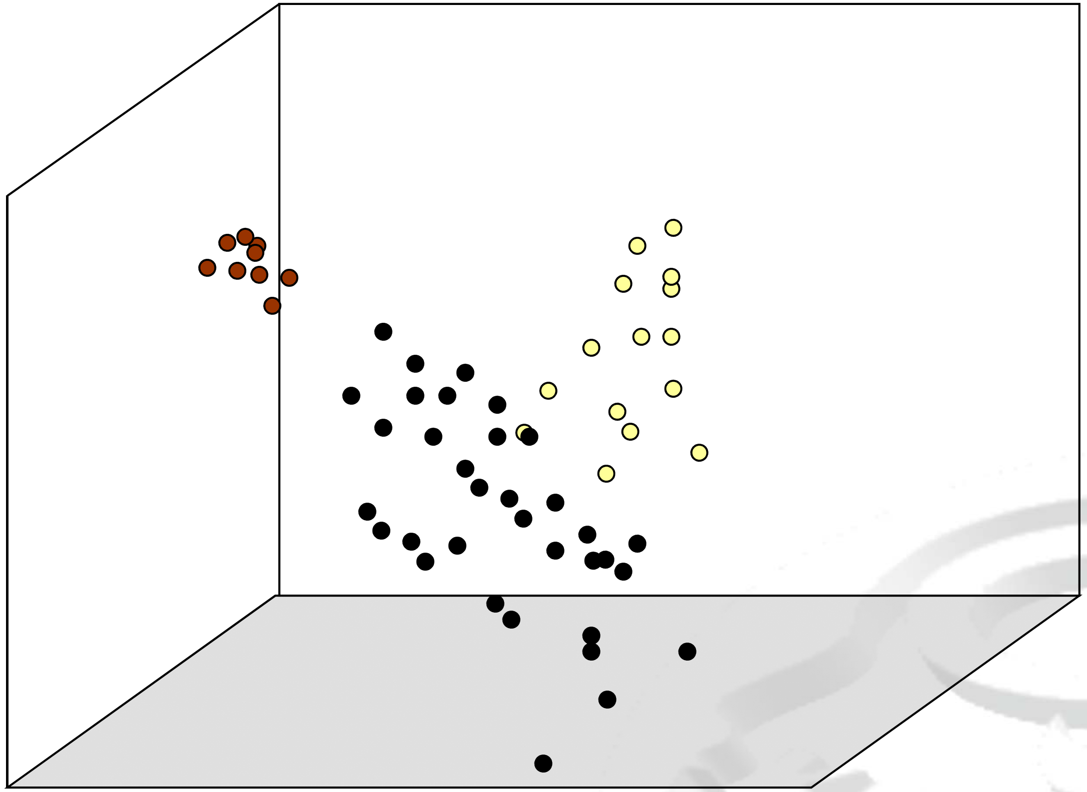


# Two-dimensional Representation



# Three-dimensional Representation



# Validation Methods for Discriminant and Predictive Models

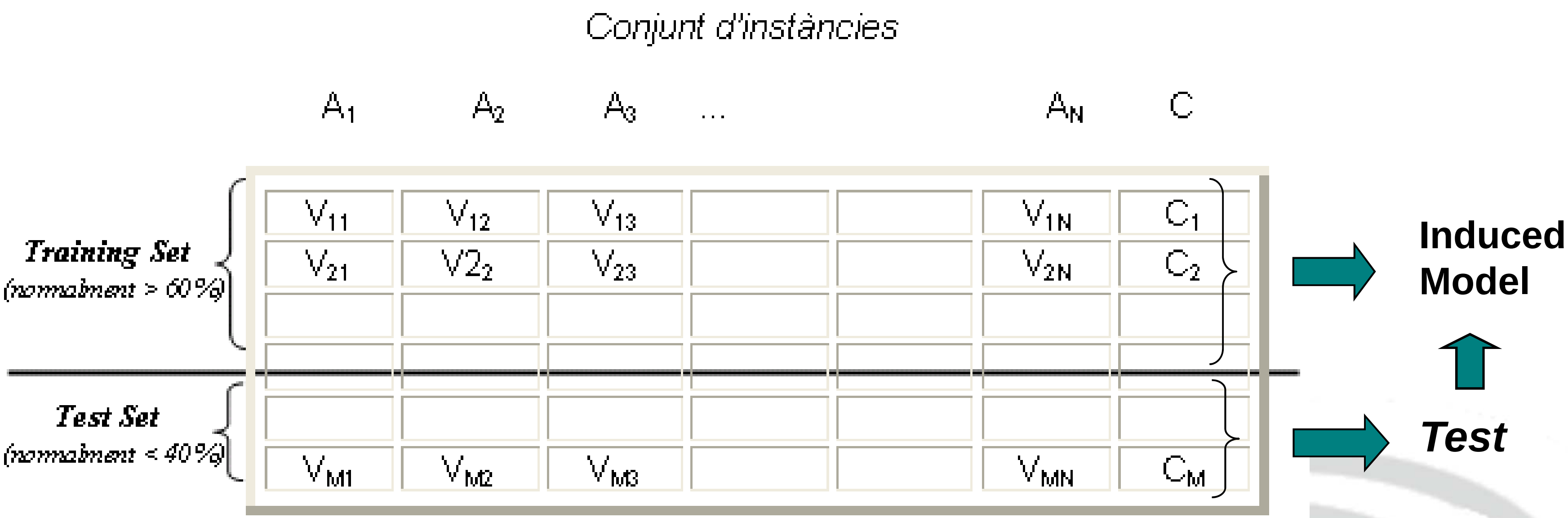
- Assessment of predictive/discriminant abilities of models
  - Training Examples
    - Obtention of the model
  - Test Examples
    - Assessment of the ***accuracy*** and ***generalization ability*** del model

- Methods and tools for rate estimation
  - Simple Validations / Cross Validations
  - Random Validations / Stratified Validations

# Precision/Accuracy Types

- Global Precision/Accuracy of classification
- Precision/Accuracy of classification by modalities
  - Confusion Matrix [General use]
  - ROC (Receiver /Relative Operating Characteristic) Curves [for binary classifiers]
    - Gini Index

# Simple Validation [Stratified] (1)




**Stratified**: The class distribution $C_1$, …,$C_M$ in the *training set* and in the *test set* follows the same distribution than the original whole data set

# Simple Validation [Stratified] (2)

**Resultats Validació**

Validació Simple (estratificada = true, percentatge = 70.0) :

ALGORISME: Rules

MATRIU DE CONFUSIÓ

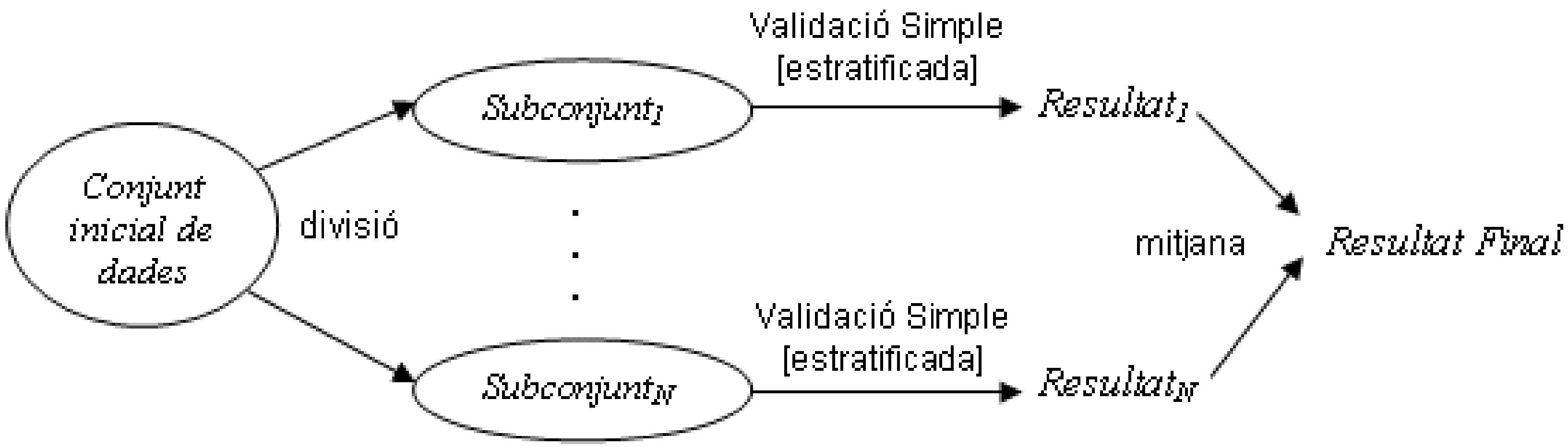
| REALS / PREDITES | Iris-versicolor | Iris-virginica | Iris-setosa | NO CLASSIFICADES | TOTALS |
|---|---|---|---|---|---|
| Iris-versicolor | 12 | 0 | 3 | 0 | 15 |
| Iris-virginica | 1 | 8 | 6 | 0 | 15 |
| Iris-setosa | 0 | 0 | 15 | 0 | 15 |
| | 13 | 8 | 24 | 0 | 45 |

PERCENTATGE CORRECTESA REGLES DE CLASSIFICACIÓ : 77.78%

# Cross Validation [Stratified]

The initial data set is split into N subsets

N = [3…10], defined by the user




The *final accuracy rate* is computed as the *average of accuracy rates* obtained for each one of the subsets

# Global Precision of classification

Global Error or Misclassification Rate

Global Accuracy or Success or Classification Rate

| MODEL | MISCLAS. RATE | VALIDATION MISCLAS. RATE | TEST MISCLAS. RATE |
|---|---|---|---|
| CART Tree | 0,2593856655 | 0,2974079127 | 0,2909836066 |
| K-NN MBR | 0,2894197952 | 0,2974079127 | 0,3155737705 |
| Regression | 0,3071672355 | 0,3383356071 | 0,3770491803 |
| RBF | 0,3051194539 | 0,3246930423 | 0,3360655738 |

# Precision by modalities (1)

- Confusion Matrix

| LOGISTIC REGRESSION | | Predicted | |
|---|---|---|---|
| | | 0 | 1 |
| Observed | 0 | 48,02 | 10,91 |
| | 1 | 22,92 | 18,14 |

Error Type II / False Positives

Error Type I / False Negatives

| CART CLASSIFICATION TREE | | Predicted | |
|---|---|---|---|
| | | 0 | 1 |
| Observed | 0 | 43,52 | 15,42 |
| | 1 | 14,32 | 26,74 |

# Precision by modalities (2)

- Confusion Matrix

| **RBF neural network** | | Predicted | |
|---|---|---|---|
| | | 0 | 1 |
| Observed | 0 | 47,34 | 11,60 |
| | 1 | 20,87 | 20,19 |

| **K-NN MBR** | | Predicted | |
|---|---|---|---|
| | | 0 | 1 |
| Observed | 0 | 41,34 | 17,60 |
| | 1 | 12,14 | 28,92 |

# ROC Curves

*Sensitivity*

True Positive rate

(1 – prob(error type I))

versus

*1- especificity*

False Positive rate

(prob(error type II)):

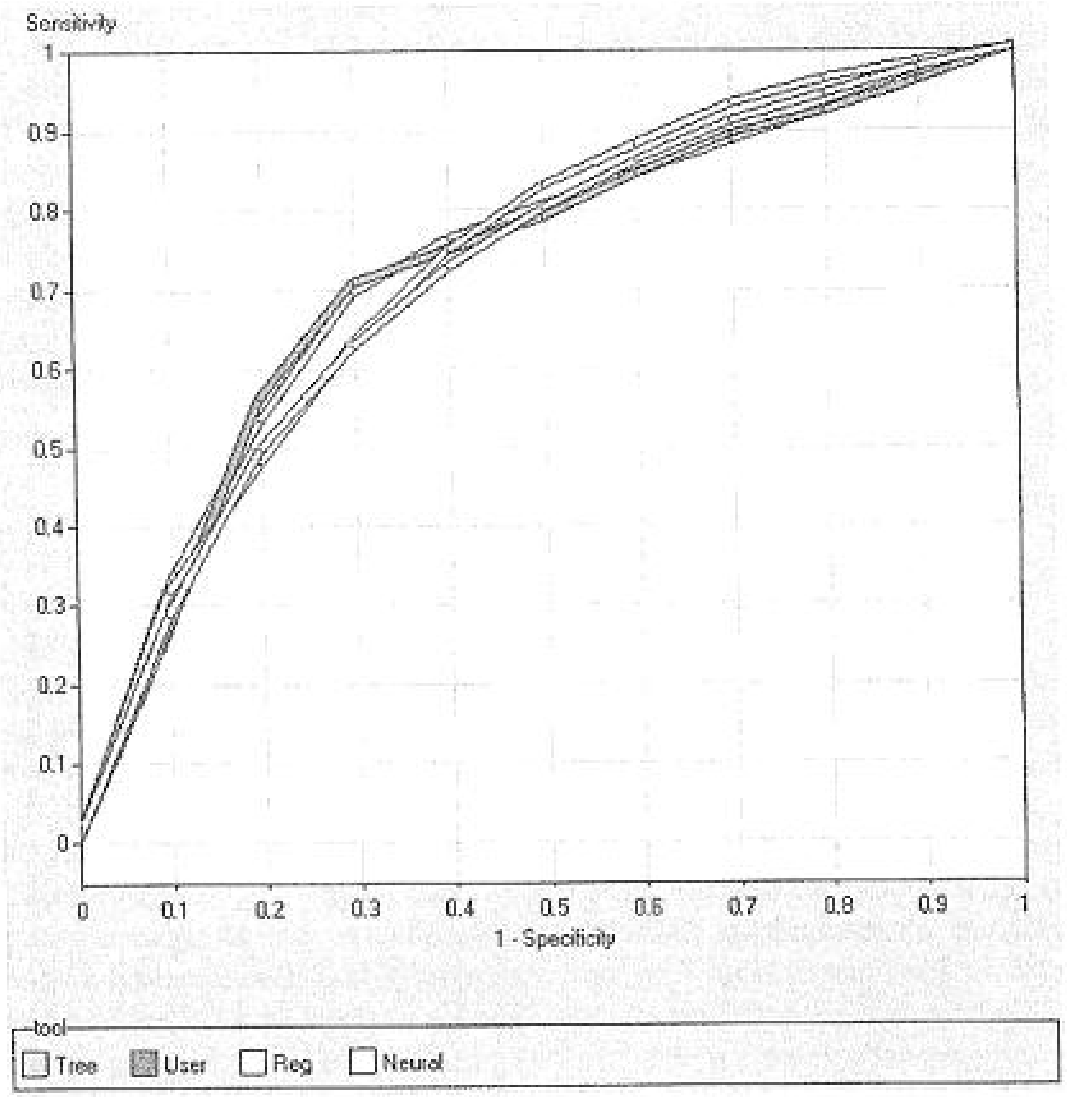



**Figure 10.5** ROC curves for the considered models. The curve called user is the MBR model.

Knowledge Engineering and Machine Learning Group
UNIVERSITAT POLITÈCNICA DE CATALUNYA

# Performance Gini Index

- Surface between ROC curve and the 45º bisector

| | Logistic Regression | RBF | CART Tree | K-NN MBR |
|---|---|---|---|---|
| **Gini index** | 0,4375 | 0,4230 | 0,4445 | 0,5673 |

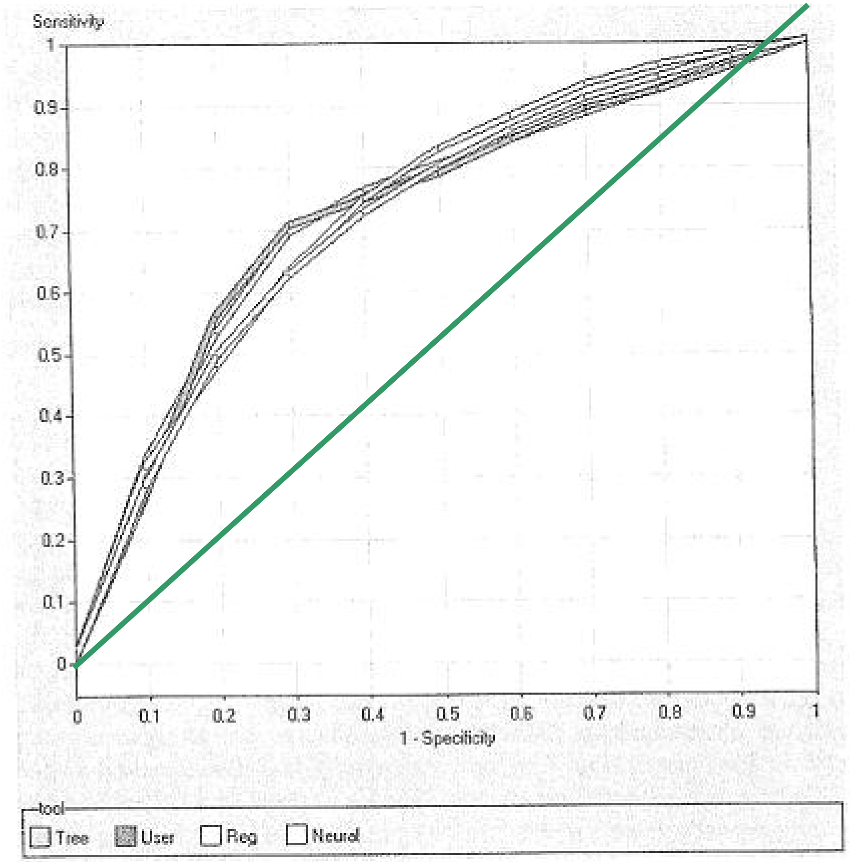



**Figure 10.5** ROC curves for the considered models. The curve called user is the MBR model.

# BIG DATA

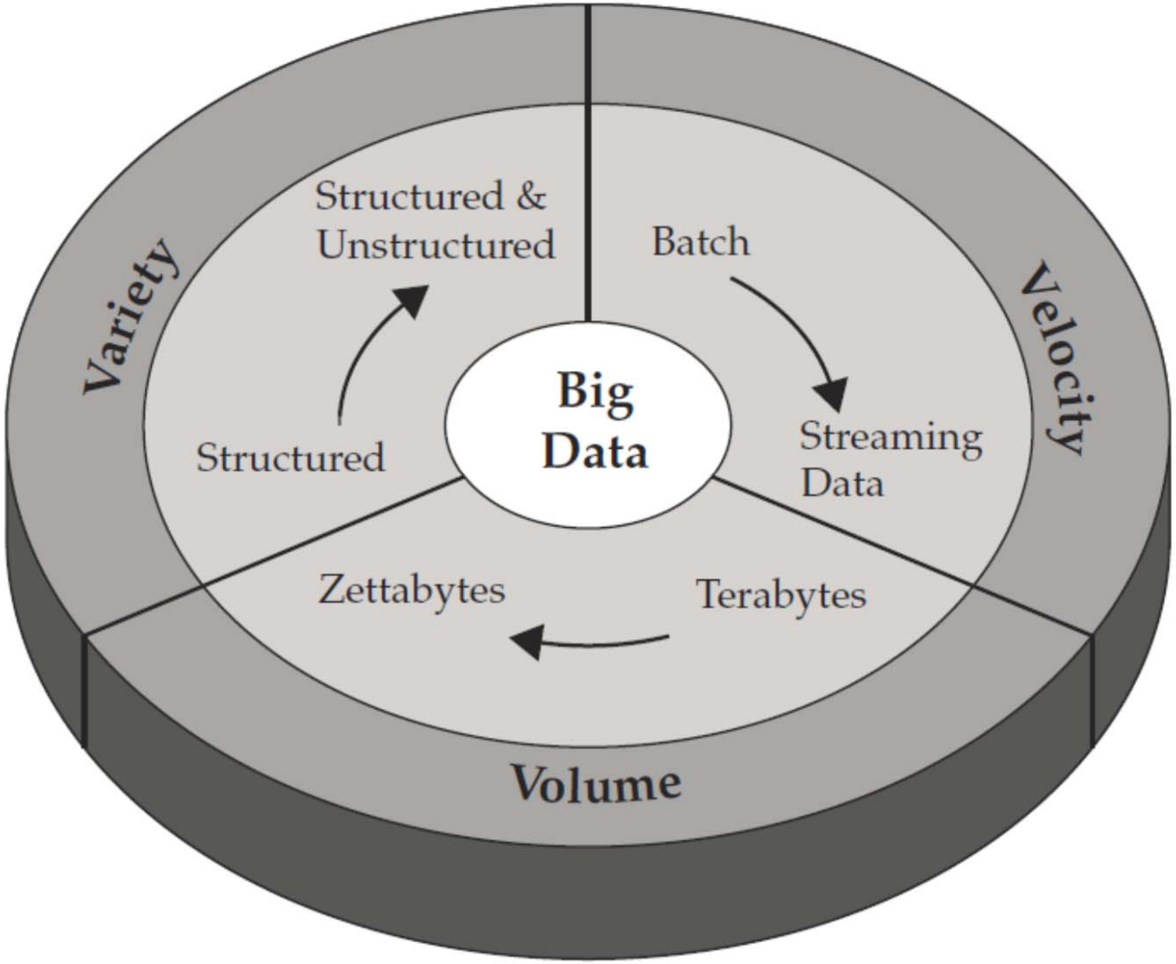
## Big data / “Small” data / Data Science / Social Data

# Big Data

- Big Data is a **collection of data sets so large and complex** that it becomes difficult to process using on-hand database management tools or traditional data processing applications.
- The **trend** to larger data sets is due to:
  - as compared to separate smaller sets with the same total amount of data
  - Increase of storage capacities
  - Increase of processing power
  - Availability of data
  - Additional information derivable from analysis of a single large set of related data
- As of 2012, limits on the size of data sets that are feasible to process in a reasonable amount of time were on the order of ***exabytes*** of data.

From DM to BD & DS: a Computational Perspective

# Big Data
[Chart from Marko Grobelnik]



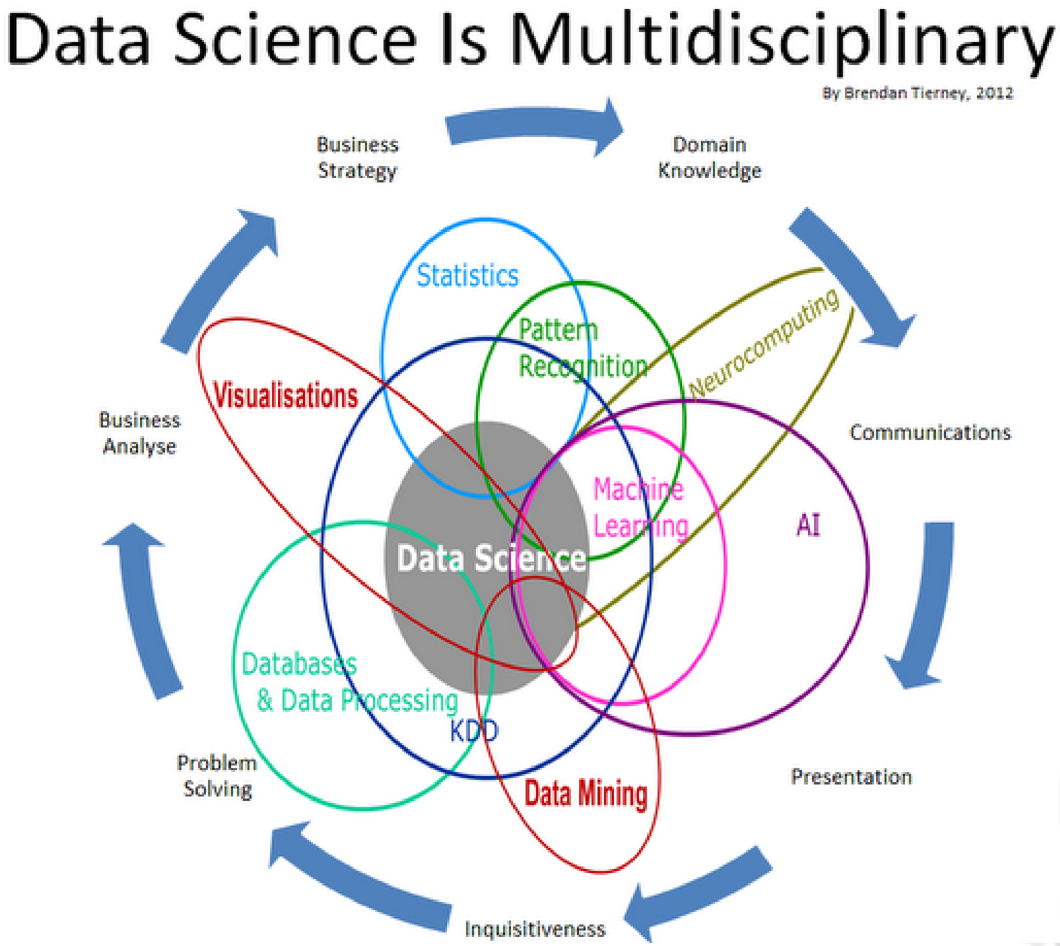
# Big Data

- Problems
  - Big data always give an answer, but it could not make sense
  - More data could imply more error
  - With enough data anything can be proved (statistics)
- Advantages
  - More tolerant to errors
  - Discover prototypical uncommon cases
  - ML algorithms work better
- Features
  - Data Volume
  - Data in Motion (streaming data)
  - Data diversity (only 20% data is related)
  - Complexity
- Philosophy: “collect first, then think”

# Big Data

- Big Data is similar to “Small Data”, but bigger ...
- Managing bigger data requires different approaches:
  - Techniques
  - Tools
  - Architectures
- The challenges include (new and old problems):
  - Capture
  - Curation
  - Storage
  - Search
  - Sharing
  - Transfer
  - **Analysis**
  - **Visualization**

# Data Science



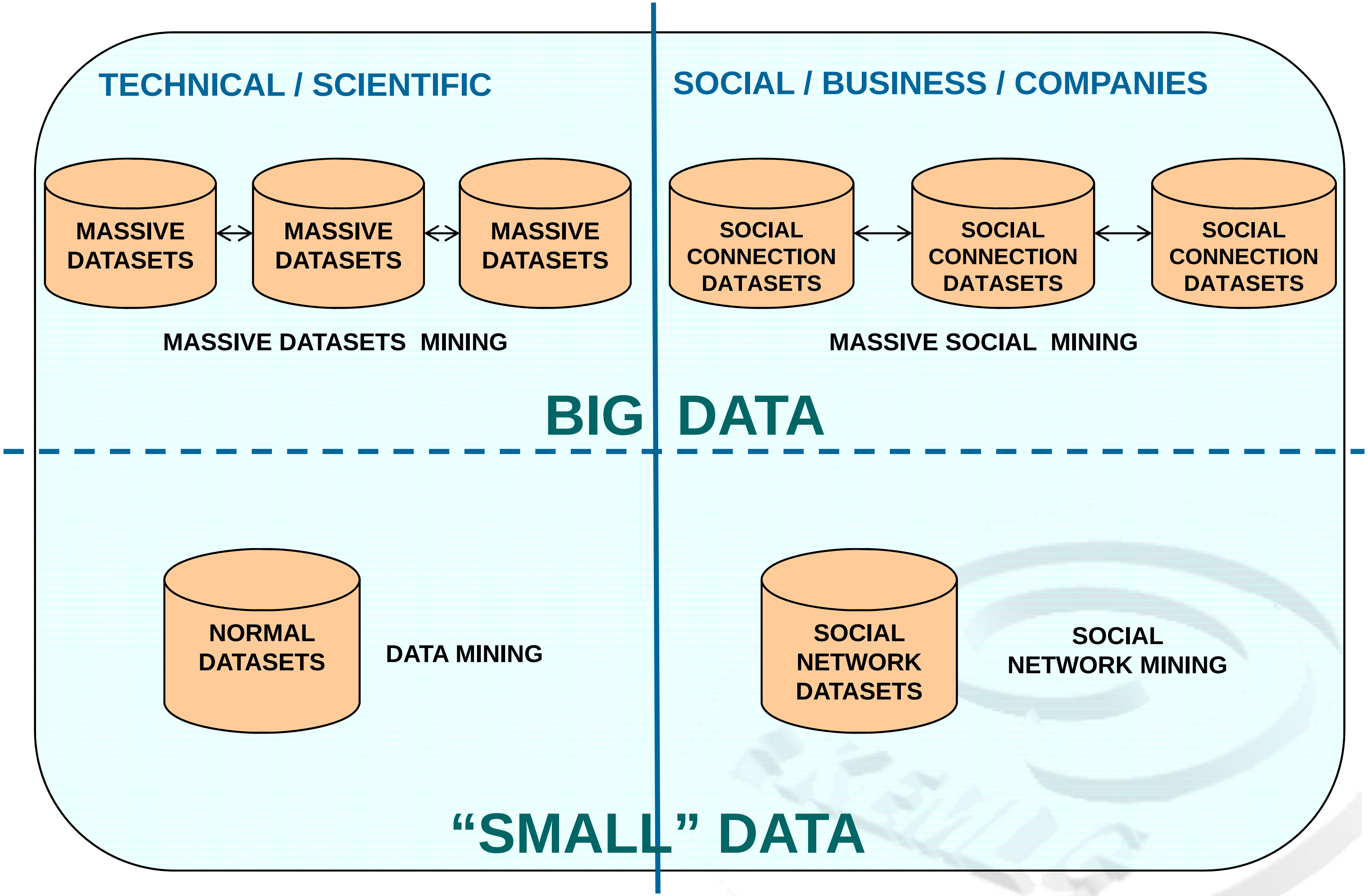

- **Data science** incorporates varying elements and builds on techniques and theories from many fields, including math, statistics, data engineering, pattern recognition and learning, advanced computing, visualization, uncertainty modeling, data warehousing, and high performance computing with the goal of **extracting meaning from data and creating data products**.

# Data Science

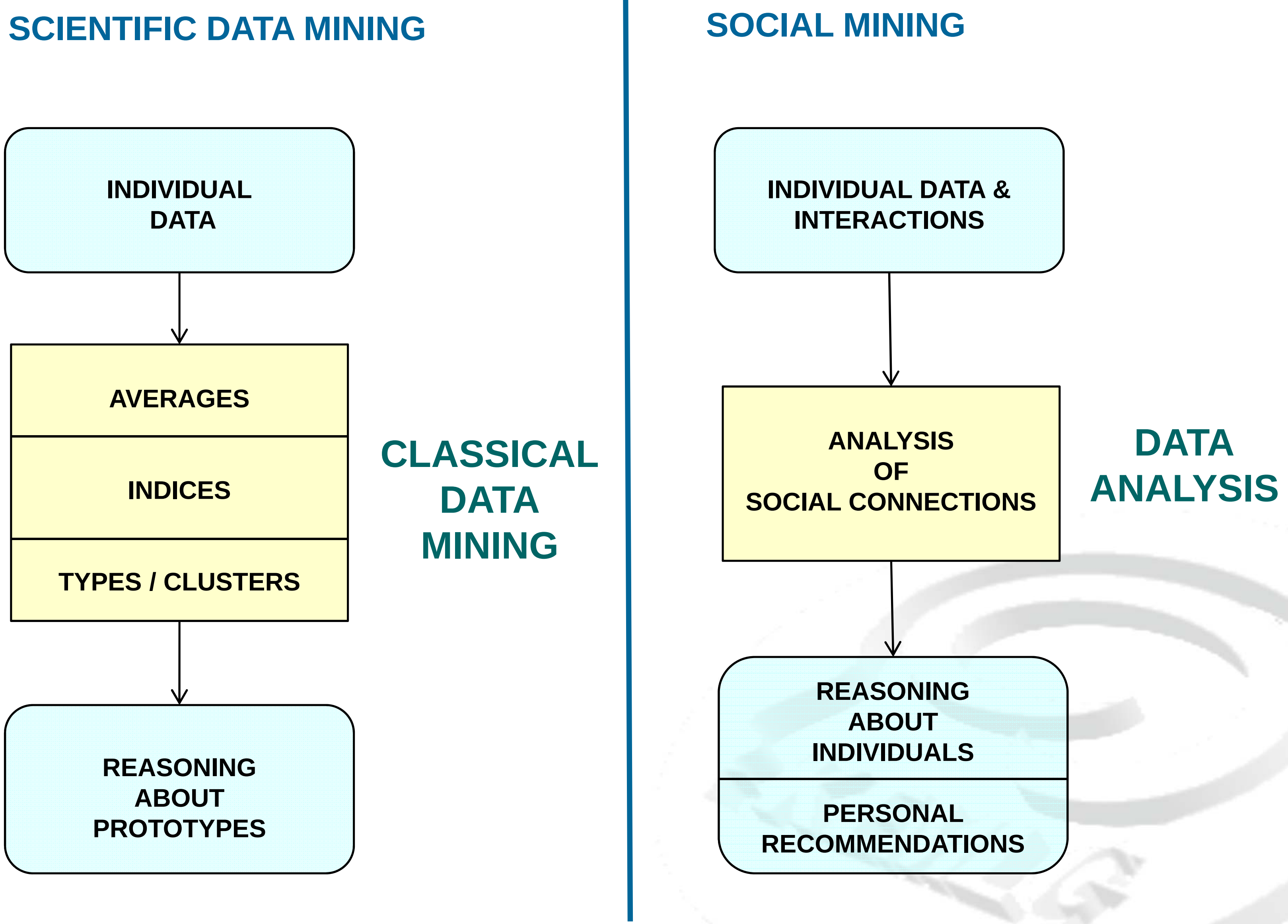
[Extracted from "Big Data Course" D. Kossmann & N. Tatbul, 2012]

- New approach to do science
  - Step 1: collect data
  - Step 2: generate hypotheses
  - Step 3: Validate Hypotheses
  - Step 4: Goto Step 1 or 2
- It can be automated: no thinking, less error
- How do you test without a ground truth?

# Big Data Jungle



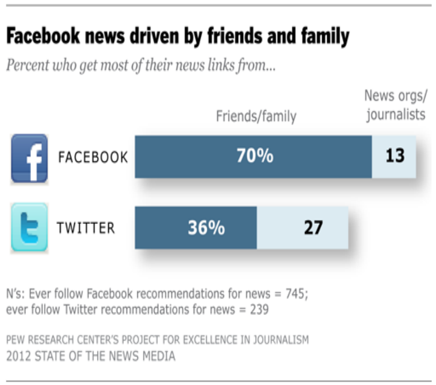
# Scientific Data Mining vs Social Mining

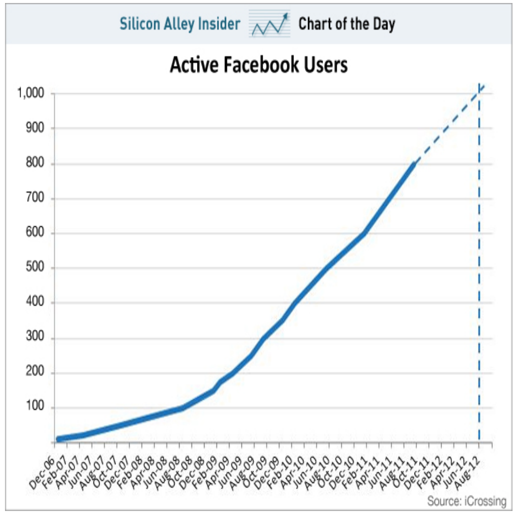


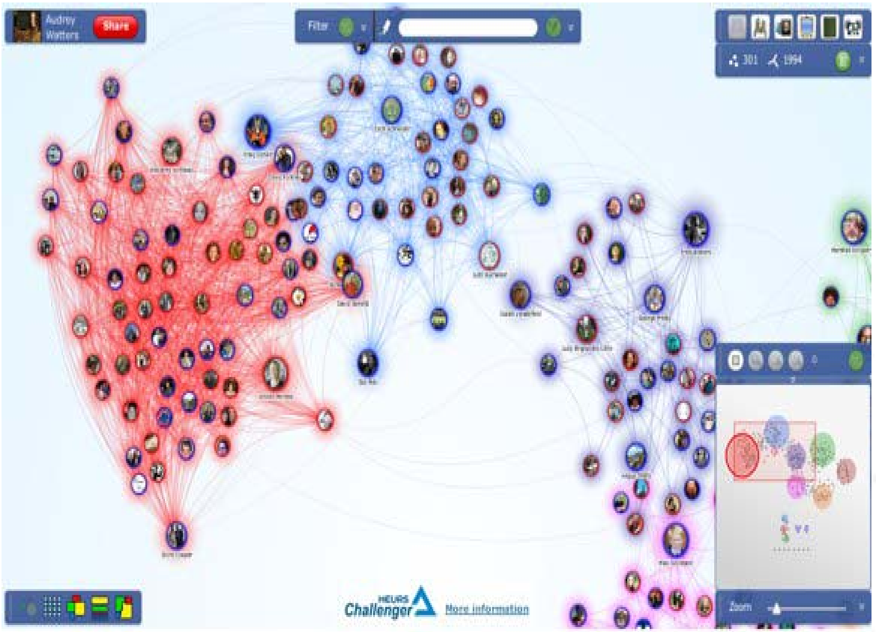
# Social Mining

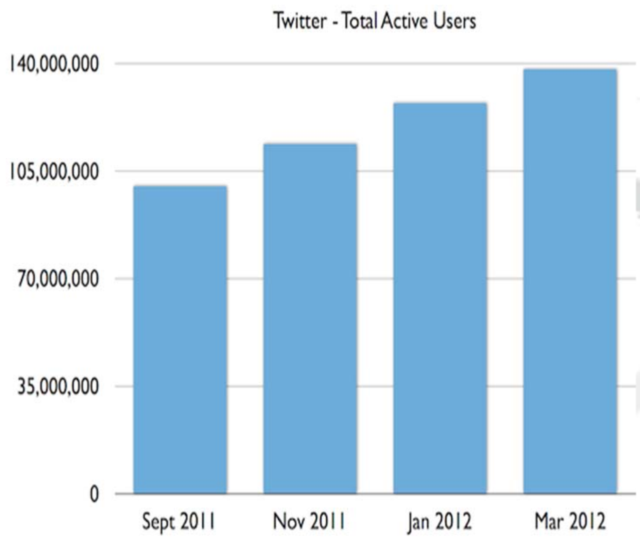
- Analyse the context and the social connections
- Analyse what you do
- Analyse where you go
- Analyse what you choose
- Analyse what you buy
- Analyse in what you spend time
- Analyse who is influencing you

# Social Netwoks Analysis and Big Data



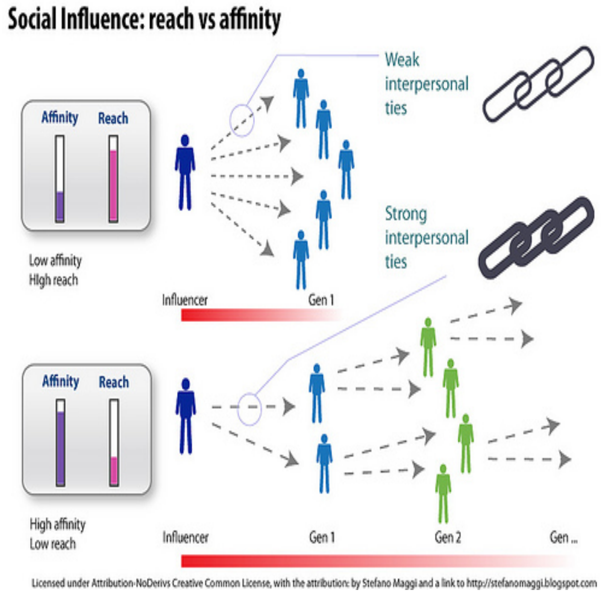



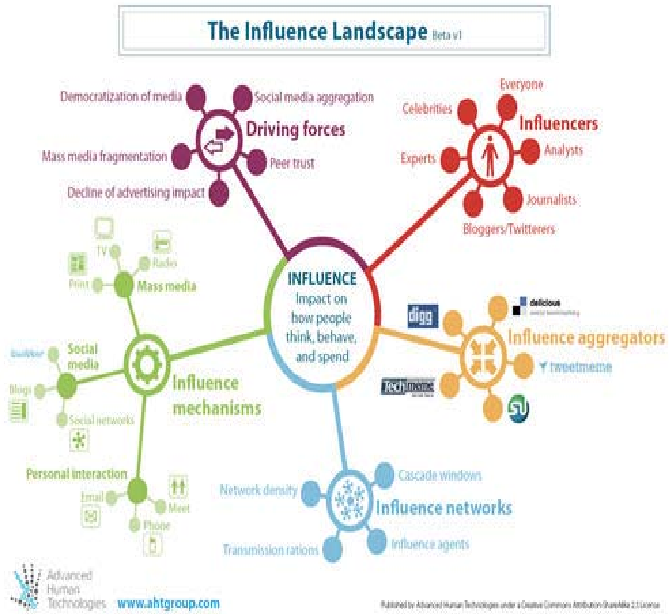



# Influence

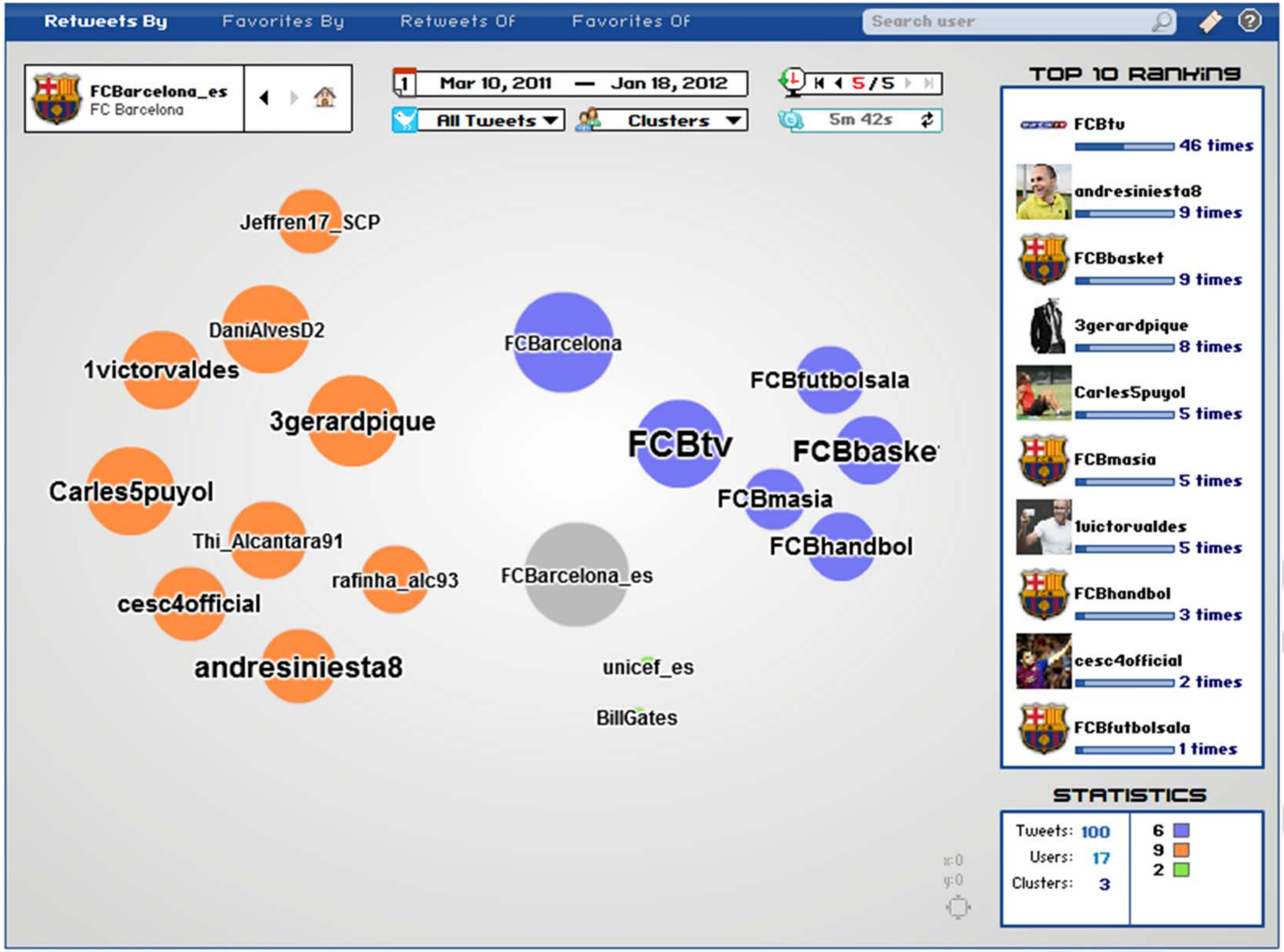

social
graph
influencers
influence
data
network
users
community
user
people
target
influencer
edges
centrality
conversation
media
time
targets
relationship
metrics
friendship
analytics
relevance
marketing
many
figure
relevant
finding
friends
sna
see
interest however
different
graphs red credible
example
connected
analysis
number
authority
also
fast
information company
iphone may
even group degree
domain
using
link


Social Influence: reach vs affinity
Weak interpersonal ties
Strong interpersonal ties
Affinity
Reach
Low affinity
High reach
Influencer
Gen 1
Affinity
Reach
High affinity
Low reach
Influencer
Gen 1
Gen 2
Gen ...
Licensed under Attribution-NoDerivs Creative Common License, with the attribution: by Stefano Maggi and a link to http://stefanomaggi.blogspot.com


The Influence Landscape
Democratization of media
Social media aggregation
Driving forces
Mass media fragmentation
Peer trust
Decline of advertising impact
Everyone
Celebrities
Influencers
Analysts
Experts
Journalists
Bloggers/Twitterers
Mass media
INFLUENCE
Impact on how people think, behave, and spend
Social media
Influence mechanisms
Influence aggregators
Personal interaction
Network density
Cascade windows
Influence networks
Influence agents
Transmission rations
Advanced Human Technologies
www.ahtgroup.com

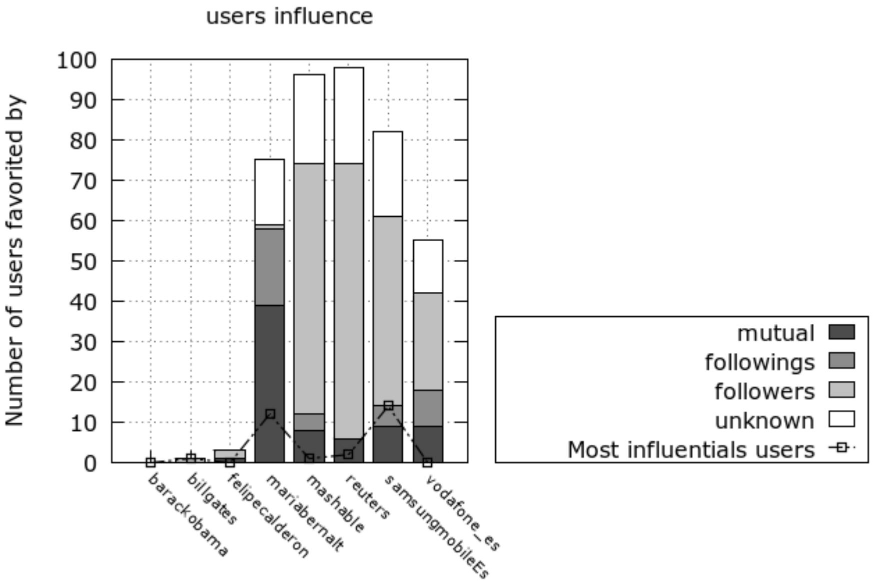
# Discovering influence based on social relationships

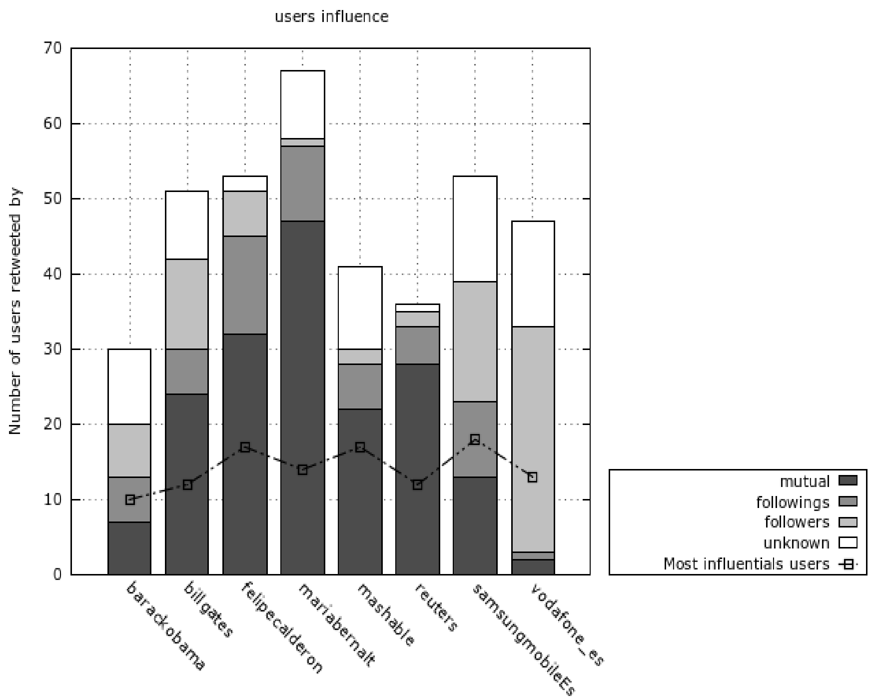
[http://www.tweetStimuli.com, A.Tejeda 2012]



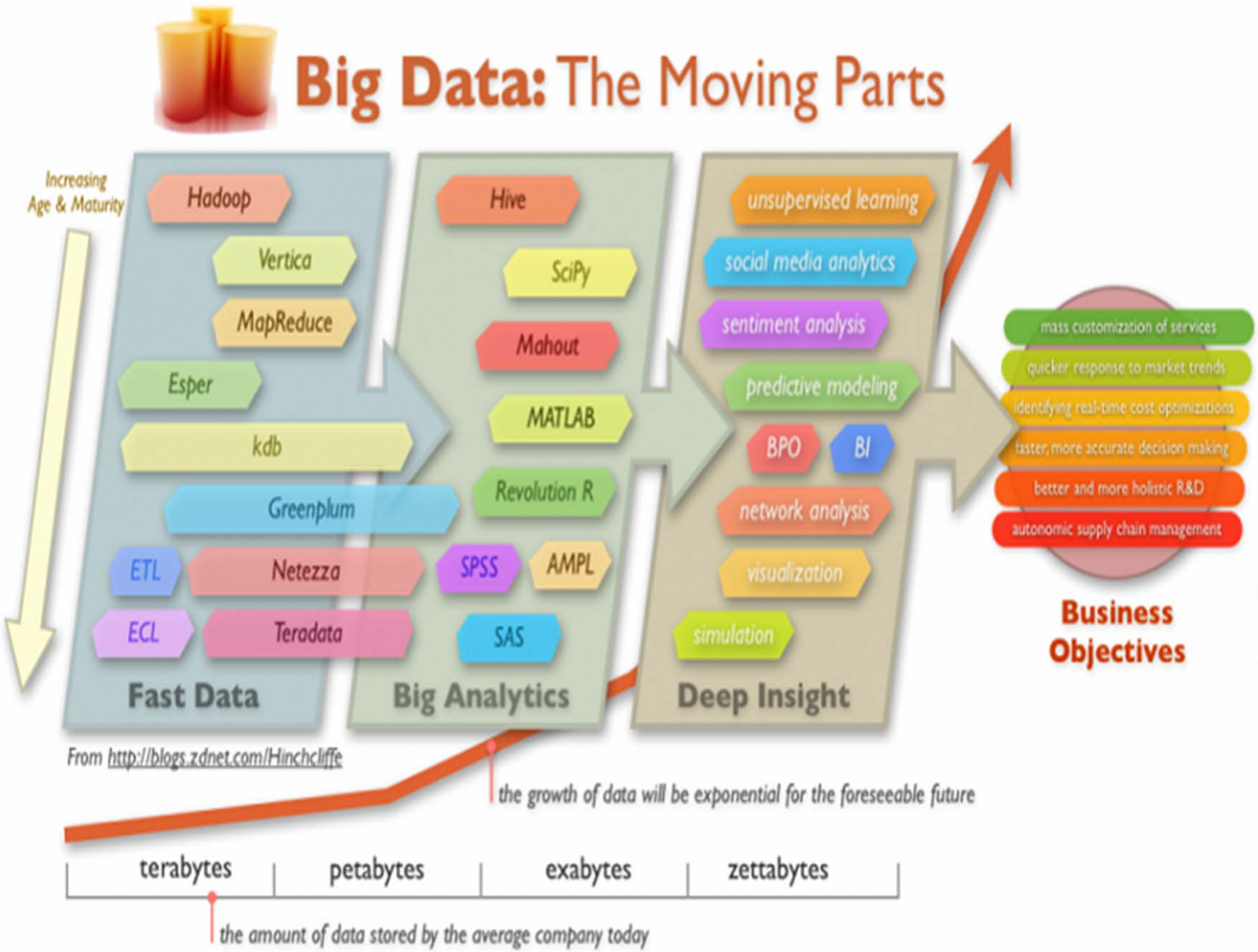
# Discovering influence based on social relationships

[http://www.tweetStimuli.com, A.Tejeda 2012]



# Social Network Analysis and Big Data

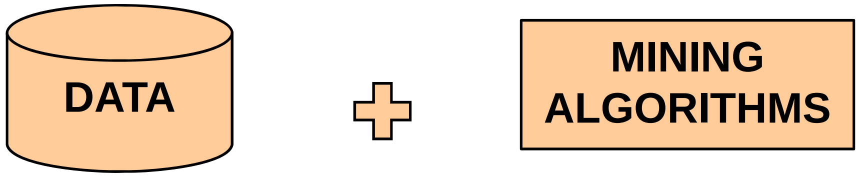


# FROM DATA MINING TO MASSIVE DATASET MINING

https://kemlg.upc.edu

Knowledge Engineering and Machine Learning Group
UNIVERSITAT POLITÈCNICA DE CATALUNYA

# How to scale from “Small” to Big Data?



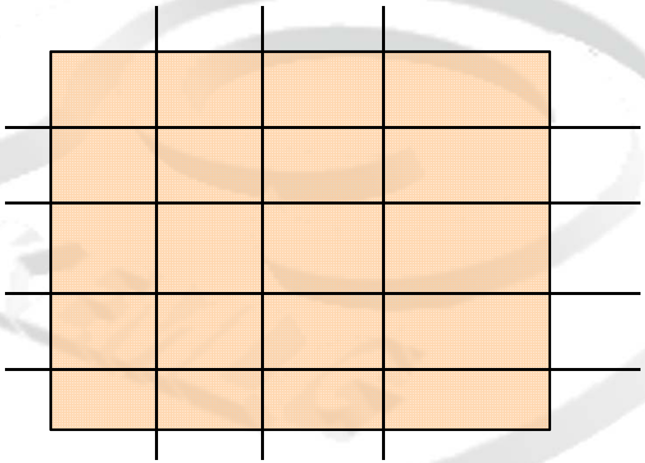

**(Distributed) High Volume or/and Slow algorithm**

- High Volume Data does not fit in Memory
  - Slow Algorithm
    - **Parallelize the algorithm + Slicing Data (Individuals or attributes)**
  - Non Slow Algorithm
    - **Slicing Data (Individuals or attributes)**
- High Volume Data Fits in Memory
  - Slow Algorithm
    - **Parallelize the algorithm**
  - Non Slow Algorithm
    - **Classical Data Mining Techniques**

# Solutions for Scaling

- Parallelizing an existing sequential algorithm
  - Exploit the instrinsic parallelism
  - ***Implies slicing data***
  - Using OpenMP, Java, C#, TBB, etc.

- Slicing Data
  - Data is splitted in ***chunks of data***
  - Each chunk is processed by **a thread** or **node of computation**
  - Slicing individuals
  - Slicing attributes
  - **Recombining the results**



# Parallel K-means Method

```
Input:   X = {x_1, ..., x_n}  // Data to be clustered
               k               // Number of clusters

Output:  C= {c_1, ..., c_k}  // Cluster centroids
```

---

```
Function K-means

Initialize C   // random selection from X
While C has changed
   For each x_i in X
      cl(x_i) = argmin_j distance (x_i, c_j)
   endfor
   For each C_j in C
      c_j = centroid ({x_i | cl(x_i) = j})
   endfor
Endwhile
return C

End function
```

**Parallelize the loop !**

**Parallelize the loop !**

# Parallel K-NN algorithm

```
Input: T = {t1, ..., tn}  // Training Data points available
       D = {d1, ..., dm}  // Data points to be classified
       k                     // Number of neighbours
Output: neighbours         // the k nearest neigbours
```

---

```
Function Parallel K-NN

Foreach data point d
   neighbours = ∅
   Foreach training point t
      dist = distance (d,t)
      If |neighbours| < k then
           insert (t, neighbours)
        else
           fartn = argmax_i distance(t,neighbours_i)
          if distance (dist < fartn)
              insert (t, neighbours)
             remove (fartn, neighbours)
          endif
`     endif
   endfor
   Return majority-vote of K-Nearest (neighbours)
endfor

End function
```

**Parallelize the loop !**

**Parallelize this loop too !**

**In high dimensions, distance computation can also benefit from parallelism !**

# Parallel Decision Tree

```
Function Parallel Attribute Selection in Decision Tree building

Max = - infinite
Foreach candidate attribute
   Relevance = quality of split(attribute)
      If (relevance > max) then
         max = relevance
         Selected attribute = attribute
      endif
endfor
return (selected attribute)
End function
```

**Parallelize the loop !**

**Some synchronization may be necessary !**

# MapReduce

[Adapted from “Big Data Course” D. Kossmann & N. Tatbul, 2012]

- A **software framework** first introduced by Google in 2004 to support parallel and fault-tolerant computations over large data sets on clusters of computers
- Based on the **map/reduce functions** commonly used in the functional programming world
- Given:
  - A very large dataset
  - A well-defined computation task to be performed on elements of this dataset (preferably, in a parallel fashion on a large cluster)
- MapReduce framework:
  - Just express what you want to compute (map() & reduce()).
  - Don’t worry about parallelization, fault tolerance, data distribution, load balancing (MapReduce takes care of these).
  - What changes from one application to another is the actual computation; the programming structure stays similar.

# MapReduce

- Here is the framework in simple terms:
  - Read lots of data.
  - **Map**: extract something that you care about from each record.
  - Shuffle and sort.
  - **Reduce**: aggregate, summarize, filter, or transform.
  - Write the results.

- One can use as many Maps and Reduces as needed to model a given problem.

# MapReduce Basic Programming Model

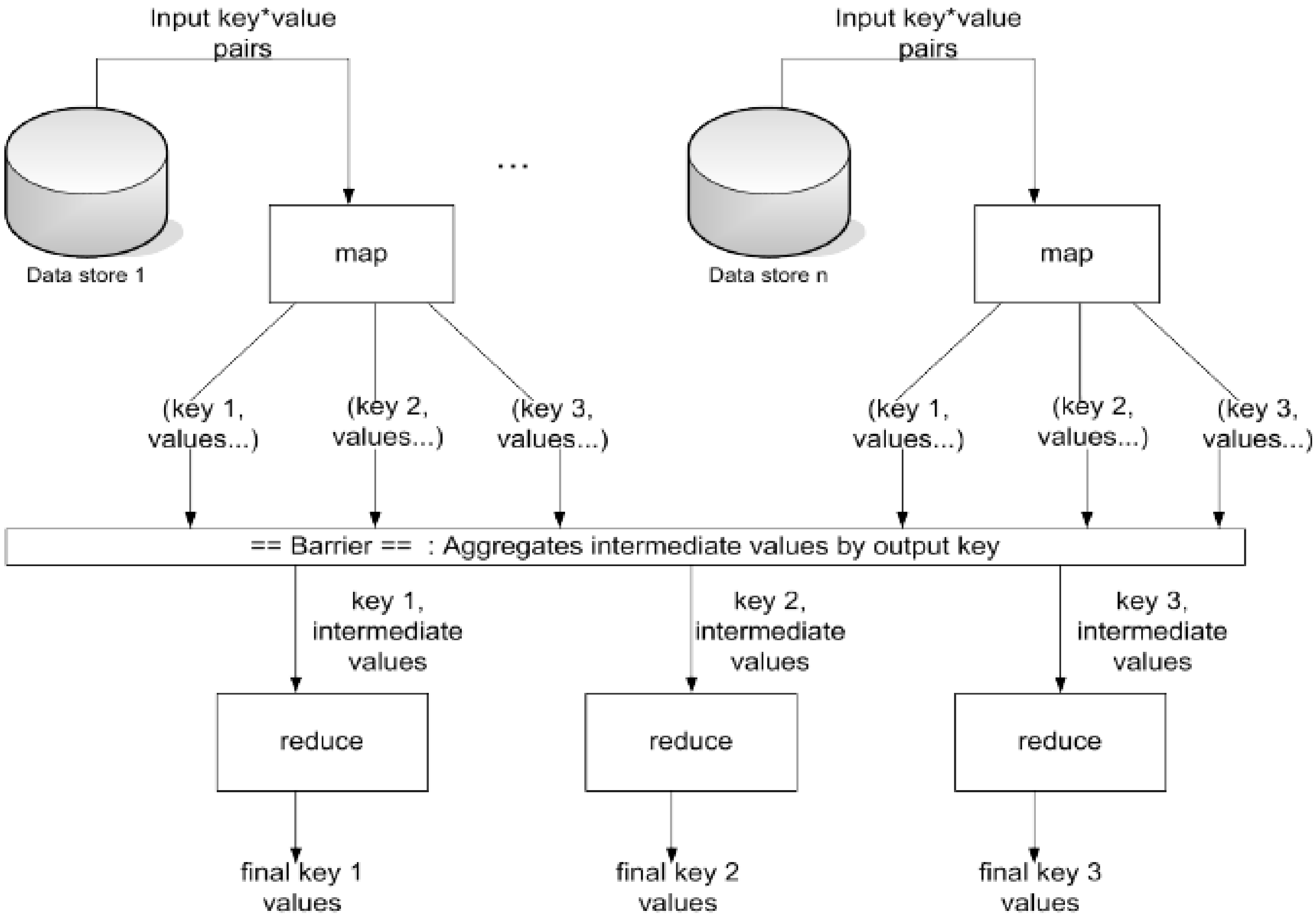
- Transform a set of input key-value pairs to a set of output values:
  - **Map**: (k1, v1) → list(k2, v2)
  - MapReduce library **groups** all intermediate pairs with same key together.
  - **Reduce**: (k2, list(v2)) → list(v2)
- Implicit parallelism in map
  - If the order of application of a function f to elements in a list is commutative, then we can reorder or parallelize execution.
  - This is the "secret" that MapReduce exploits.

# MapReduce Parallelization

- Multiple **map() functions run in parallel**, creating different intermediate values from different input data sets.
- Multiple **reduce() functions also run in parallel**, each working on a different output key.
- All values are processed independently.
- *Bottleneck: The reduce phase can't start until the map phase is completely finished.*

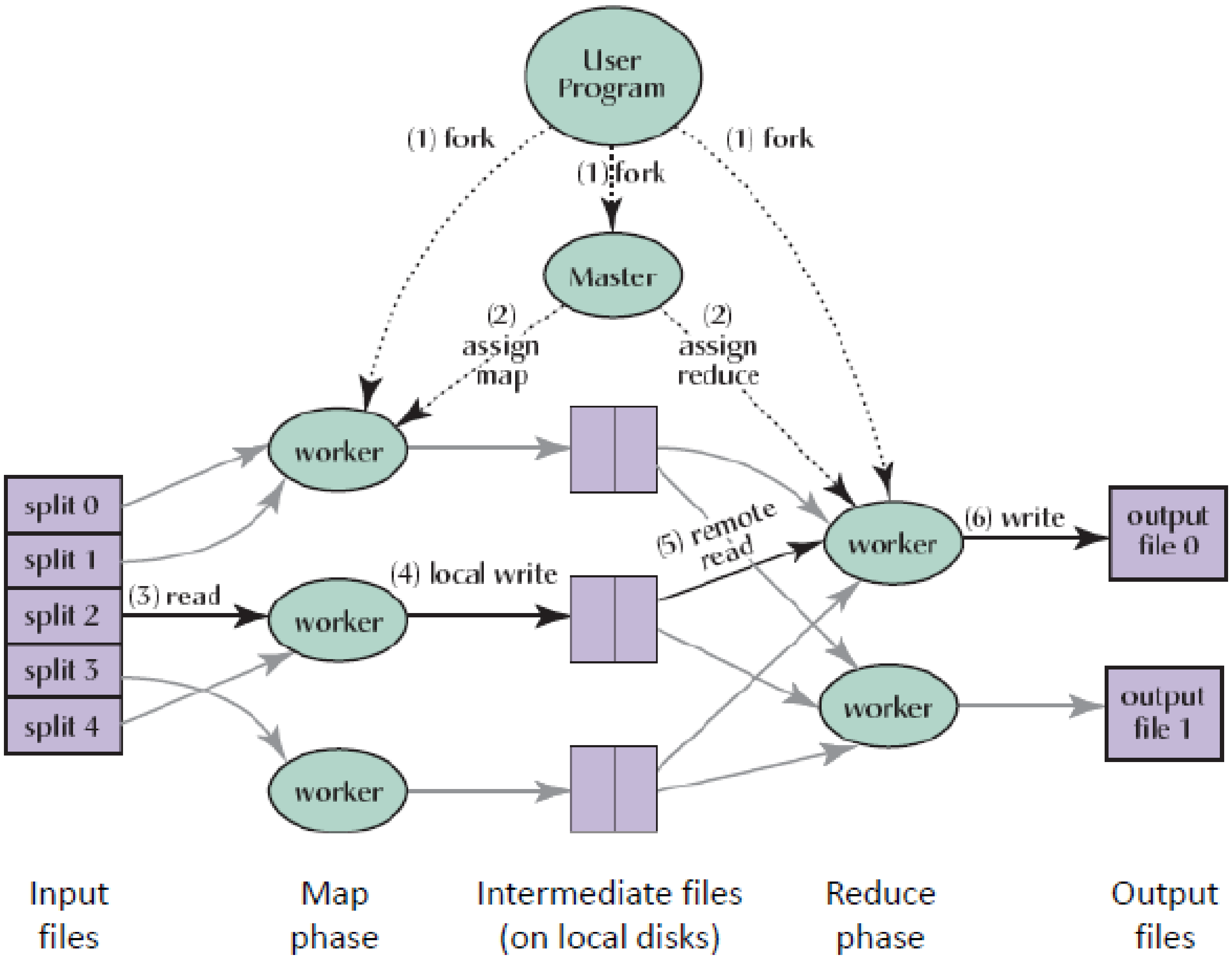
# MapReduce Parallel Processing Model

[Chart from “Big Data Course” D. Kossmann & N. Tatbul, 2012]



# MapReduce Execution Overview

[Chart from “Big Data Course” D. Kossmann & N. Tatbul, 2012]



# K-means with MapReduce

```
X = {x_1, ..., x_n}  // Data to be clustered
          k          // Number of clusters

Output:  C= {c_1, ..., c_k}      // Cluster centroids
```

| Map | Grouping | Reduce | |
|---|---|---|---|
| $(x_1, ?)$ | $(x_1, cl(x_1))$ | $(All\ cl(x_i)=1, x_i)$ | $(cl(x_i)=1, c_1= \text{average}\ (x_i \mid cl(x_i) = 1))$ |
| .... | ... | .... | ..... |
| $(x_n, ?)$ | $(x_n, cl(x_n))$ | $(All\ cl(x_i)=k, x_j)$ | $(cl(x_i)=k, c_1= \text{average}\ (x_i \mid cl(x_i) = k))$ |

# A look to MAHOUT

# Mahout

- Mahout is an open source machine learning library from Apache
- The algorithms implemented are from the **ML field**. By the moment they are:
  - Collaborative filtering/recomender engines
  - Clustering
  - Classification
- It is scalable. Implemented in Java, and some code upon Apache's Hadoop distributed computation.
- It is a Java Library
- Mahout started at 2008, as a subproject of Apache Lucene's project

# BIG DATA TOOLS & RESOURCES

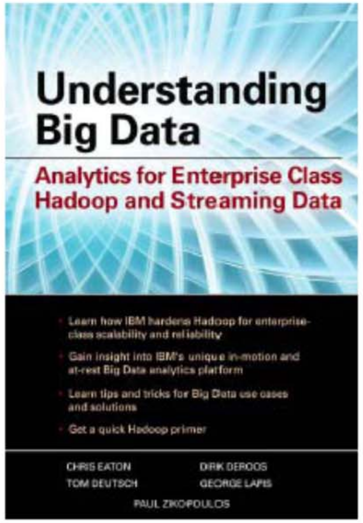
https://kemlg.upc.edu

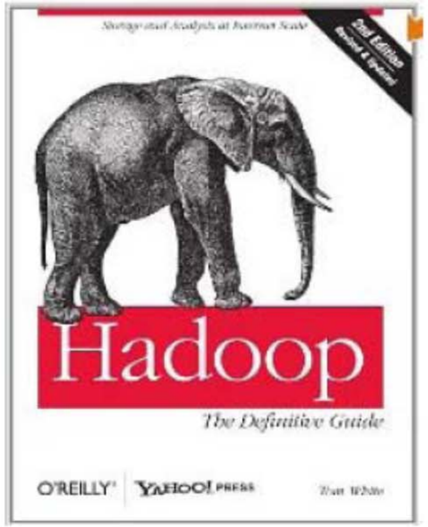
Knowledge Engineering and Machine Learning Group
UNIVERSITAT POLITÈCNICA DE CATALUNYA

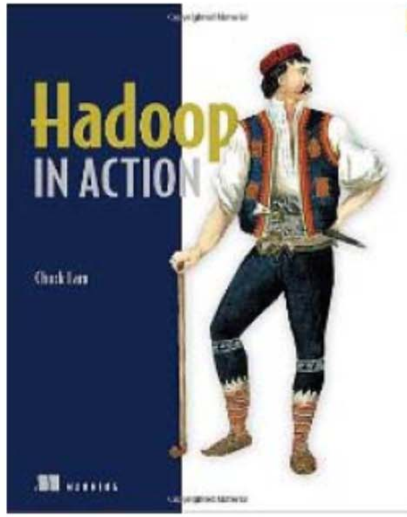
# Big Data Tools

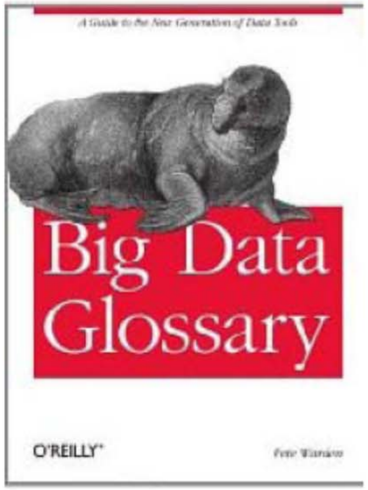
- NoSQL
  - DatabasesMongoDB, CouchDB, Cassandra, Redis, BigTable, Hbase, Hypertable, Voldemort, Riak, ZooKeeper
- MapReduce
  - Hadoop, Hive, Pig, Cascading, Cascalog, mrjob, Caffeine, S4, MapR, Acunu, Flume, Kafka, Azkaban, Oozie, Greenplum
- Storage
  - S3, Hadoop Distributed File System
- Servers
  - EC2, Google App Engine, Elastic, Beanstalk, Heroku
- Processing
  - R, Yahoo! Pipes, Mechanical Turk, Solr/Lucene, ElasticSearch, Datameer, BigSheets, Tinkerpop

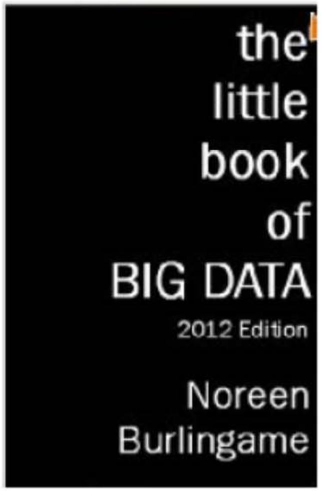
# Big Data Literature (1)

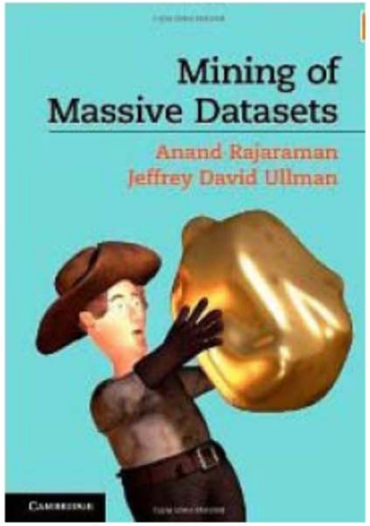



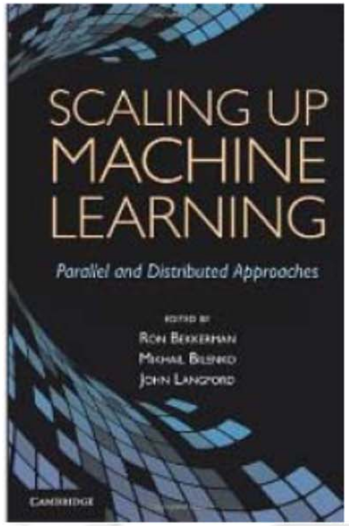



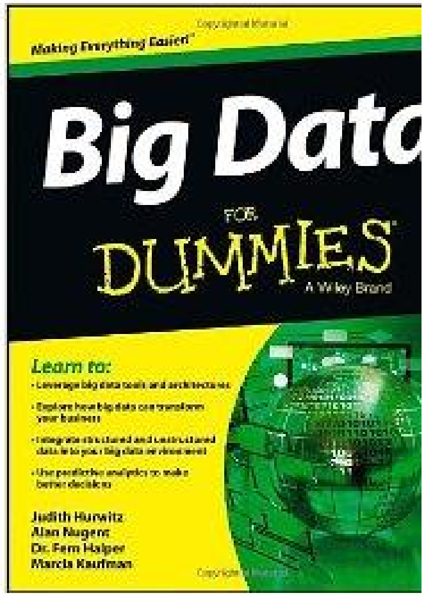



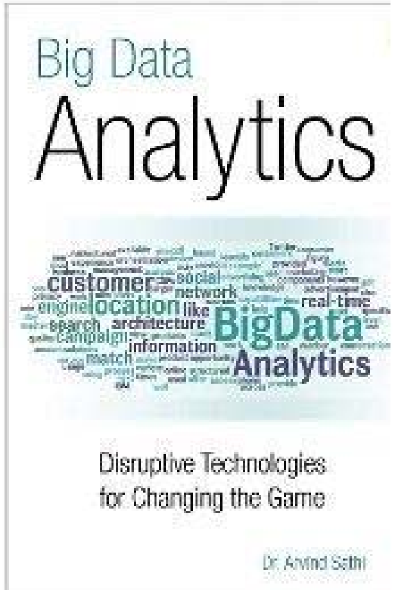



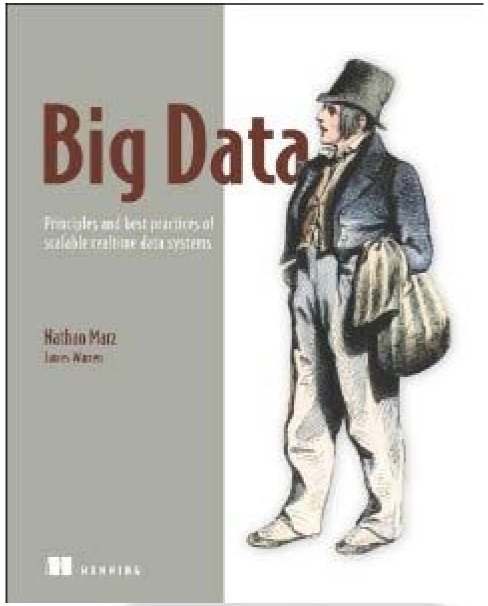



# Big Data Literature (2)



# Big Data websites

- http://www.DataScienceCentral.com
- http://www.apache.org
- http://hadoop.apache.org
- http://mahout.apache.org
- http://bigml.com

**Miquel Sànchez i Marrè**
**(miquel@lsi.upc.edu)**

**http://kemlg.upc.edu/**