BDML
Big Data meets Machine Learning,
Summer Training Workshop, Wroclaw, 2015

Big Data meets Machine Learning 2015

Summer Training Workshop in Wroclaw, Poland

May 21-24 2015

Data Preprocessing

Friday 22, 13:30

Francisco Herrera

Research Group on Soft Computing and Information Intelligent Systems (SCI²S)
http://sci2s.ugr.es
Dept. of Computer Science and A.I.
University of Granada, Spain

Email: herrera@decsai.ugr.es

DECSAI
Universidad de Granada

Motivation

Data Preprocessing: Tasks to discover quality data prior to the use of knowledge extraction algorithms.

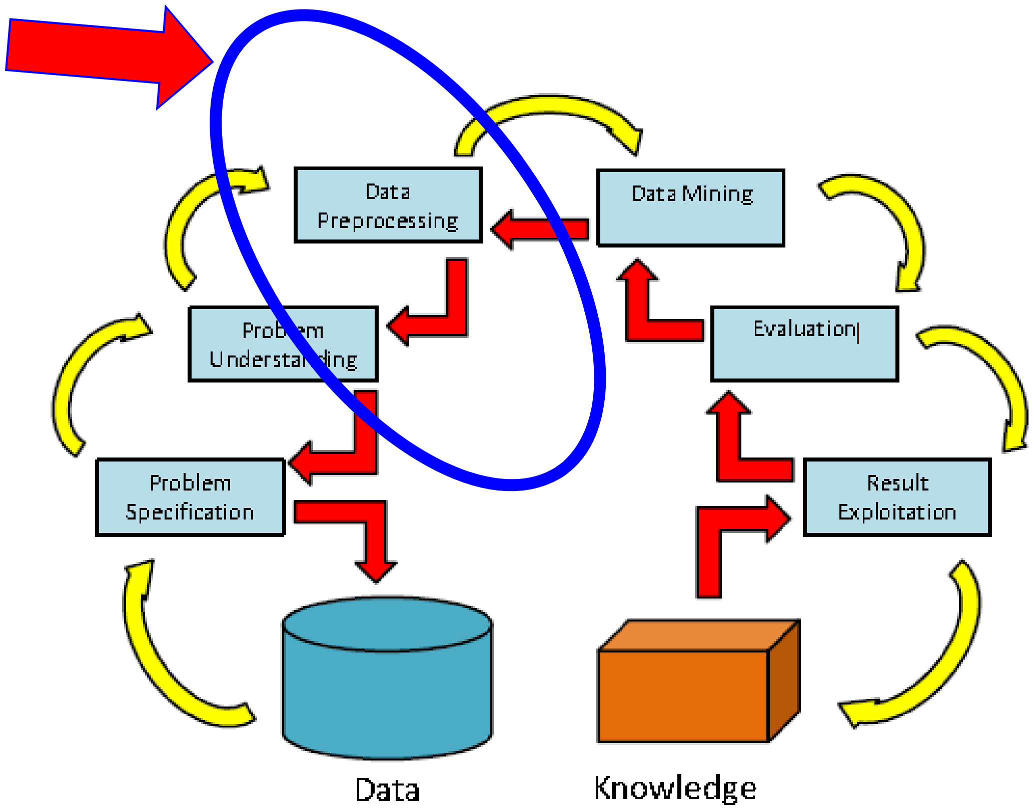
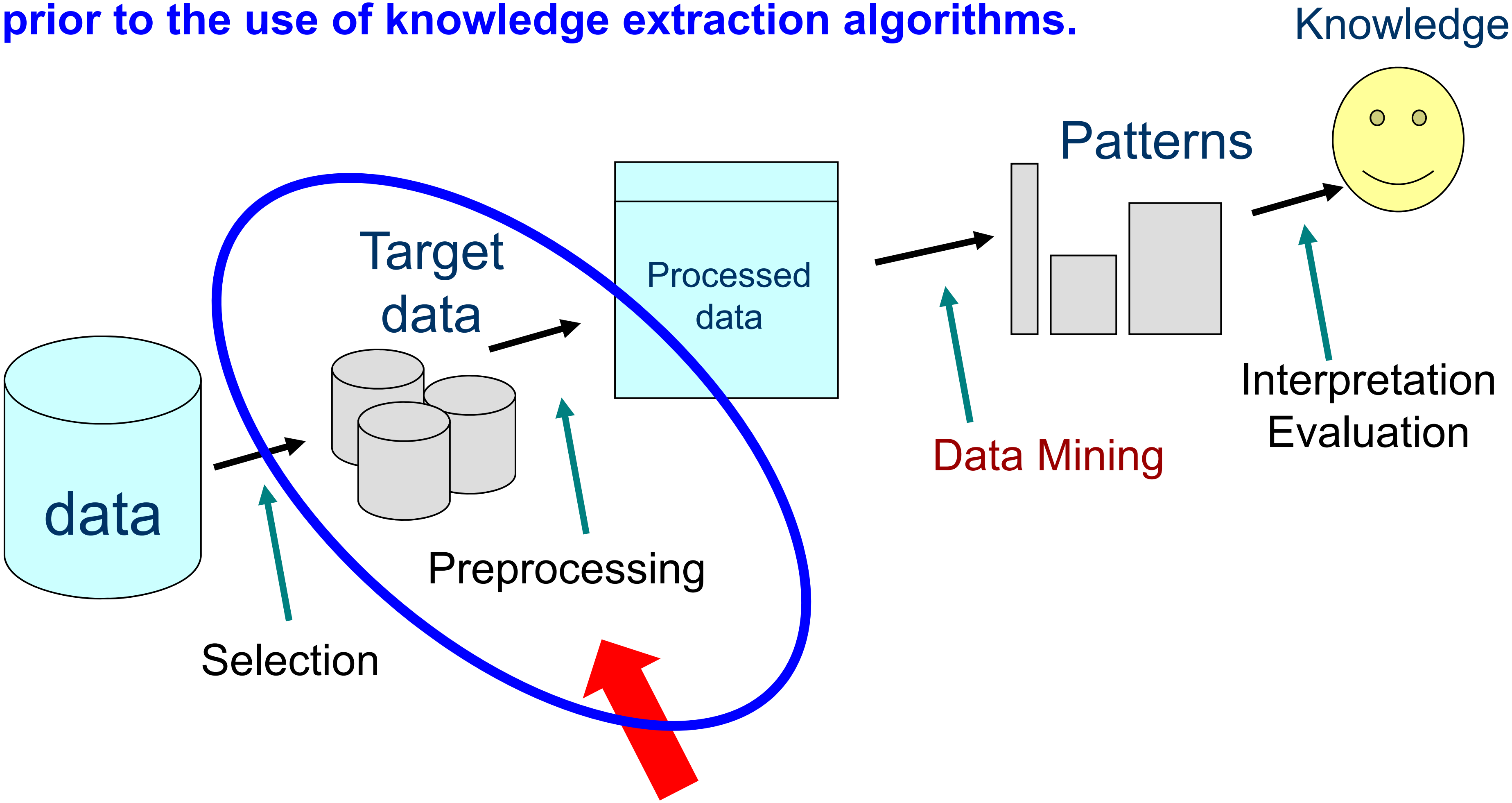


Fig. 1.1: KDD process.

Motivation

Data Preprocessing: Tasks to <u>discover quality data</u> prior to the use of knowledge extraction algorithms.

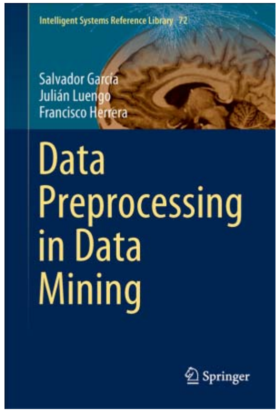


Objectives

- To understand the different problems to solve in the processes of data preprocessing.
- To know the problems in the data integration from different sources and sets of techniques to solve them.
- To know the problems related to clean data and to mitigate imperfect data, together with some techniques to solve them.
- To understand the necessity of applying data transformation techniques.
- To know the data reduction techniques and the necessity of their application.

Data Preprocessing

1. Introduction. Data Preprocessing
2. Integration, Cleaning and Transformations
3. Imperfect Data
4. Data Reduction
5. Final Remarks



Bibliography:

S. García, J. Luengo, F. Herrera
Data Preprocessing in Data Mining
Springer, Enero 2015

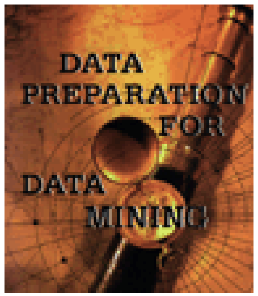
Data Preprocessing in Data Mining

1. Introduction. Data Preprocessing
2. Integration, Cleaning and Transformations
3. Imperfect Data
4. Data Reduction
5. Final Remarks

INTRODUCTION

D. Pyle, 1999, pp. 90:

“The fundamental purpose of data preparation is to manipulate and transforrm raw data so that the information content enfolded in the data set can be exposed, or made more easily accesible.”



Dorian Pyle
Data Preparation for Data Mining Morgan Kaufmann Publishers, 1999

Data Preprocessing

Importance of Data Preprocessing

1. Real data could be dirty and could drive to the extraction of useless patterns/rules.

   This is mainly due to:

   Incomplete data: lacking attribute values, …
   Data with noise: containing errors or outliers
   Inconsistent data (including discrepancies)

Data Preprocessing

Importance of Data Preprocessing

2. Data preprocessing can generate a smaller data set than the original, which allows us to improve the efficiency in the Data Mining process.

   This performing includes Data Reduction techniques:
   Feature selection, sampling or instance selection, discretization.

Data Preprocessing

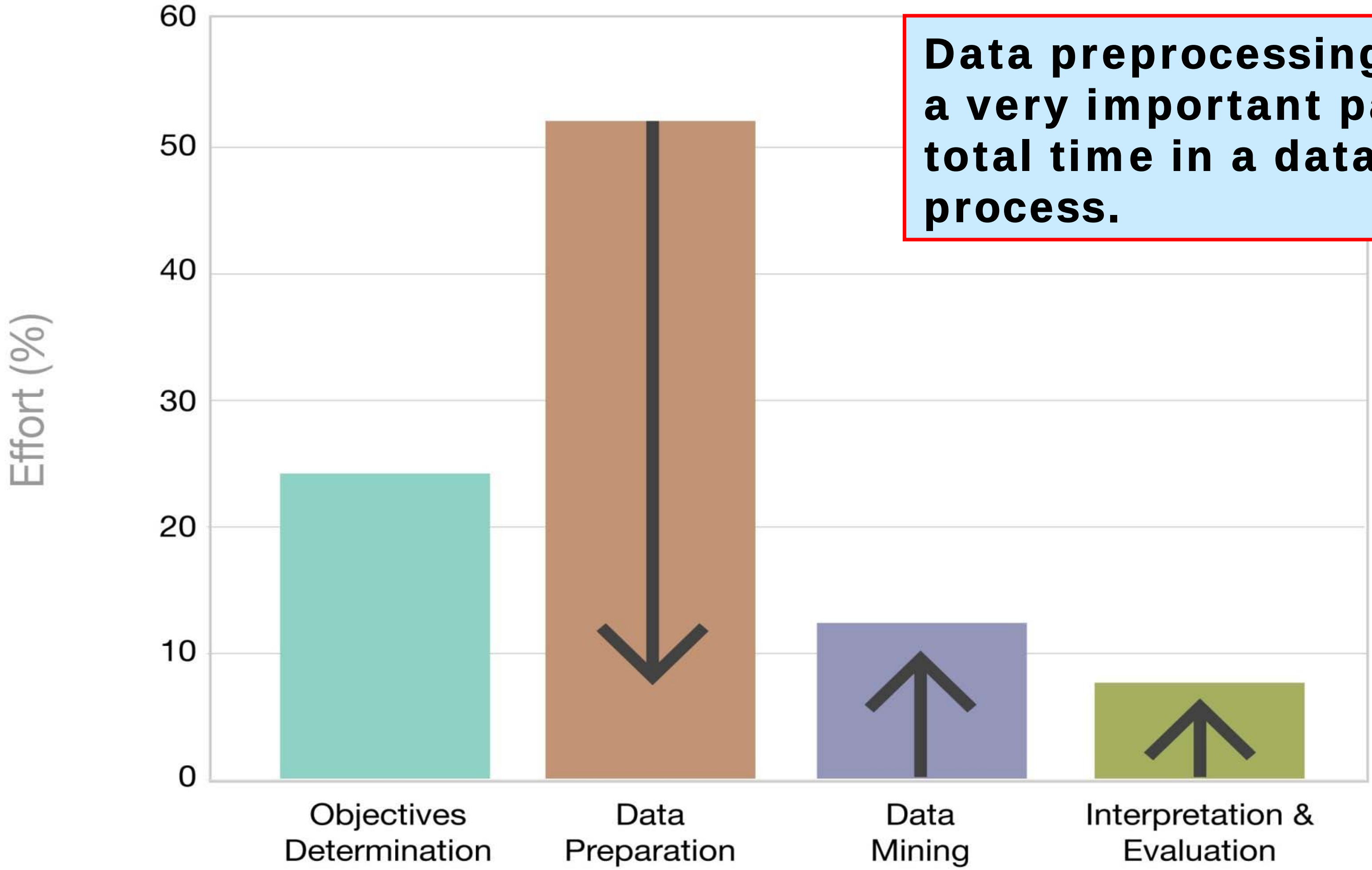
Importance of Data Preprocessing

3. No quality data, no quality mining results!

   Data preprocessing techniques generate "quality data", driving us to obtain "quality patterns/rules".

Quality decisions must be based on quality data!

Data Preprocessing



Data preprocessing spends a very important part of the total time in a data mining process.

Data Preprocessing

What is included in data preprocessing?

Real databases usually contain noisy data, missing data, and inconsistent data, …

Major Tasks in Data Preprocessing

1. Data integration. Fusion of multiple sources in a Data Warehousing.
2. Data cleaning. Removal of noise and inconsistencies.
3. Missing values imputation.
4. Data Transformation.
5. Data reduction.

Data Preprocessing

What is included in data preprocessing?

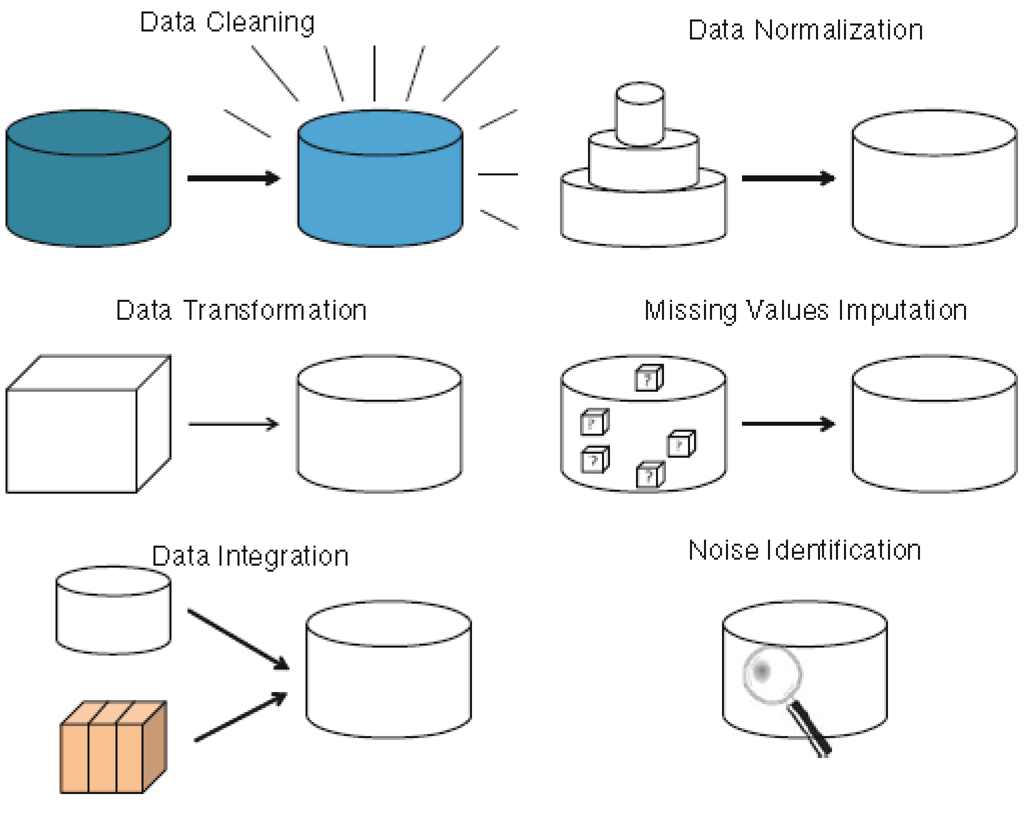


Fig. 1.3 Forms of data preparation

Data Preprocessing

What is included in data preprocessing?

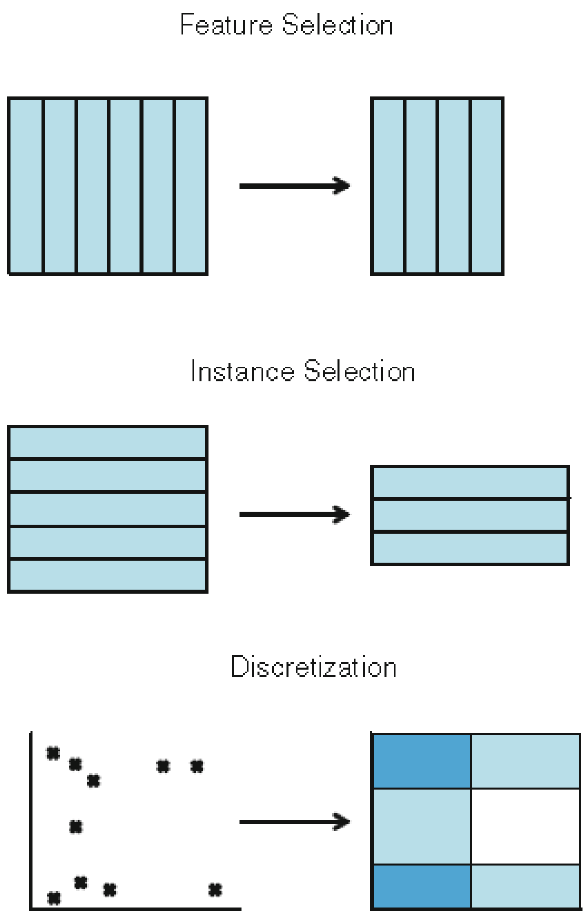


Fig. 1.4 Forms of data reduction

Data Preprocessing in Data Mining

1. Introduction. Data Preprocessing
2. Integration, Cleaning and Transformations
3. Imperfect Data
4. Data Reduction
5. Final Remarks

Integration, Cleaning and Transformation

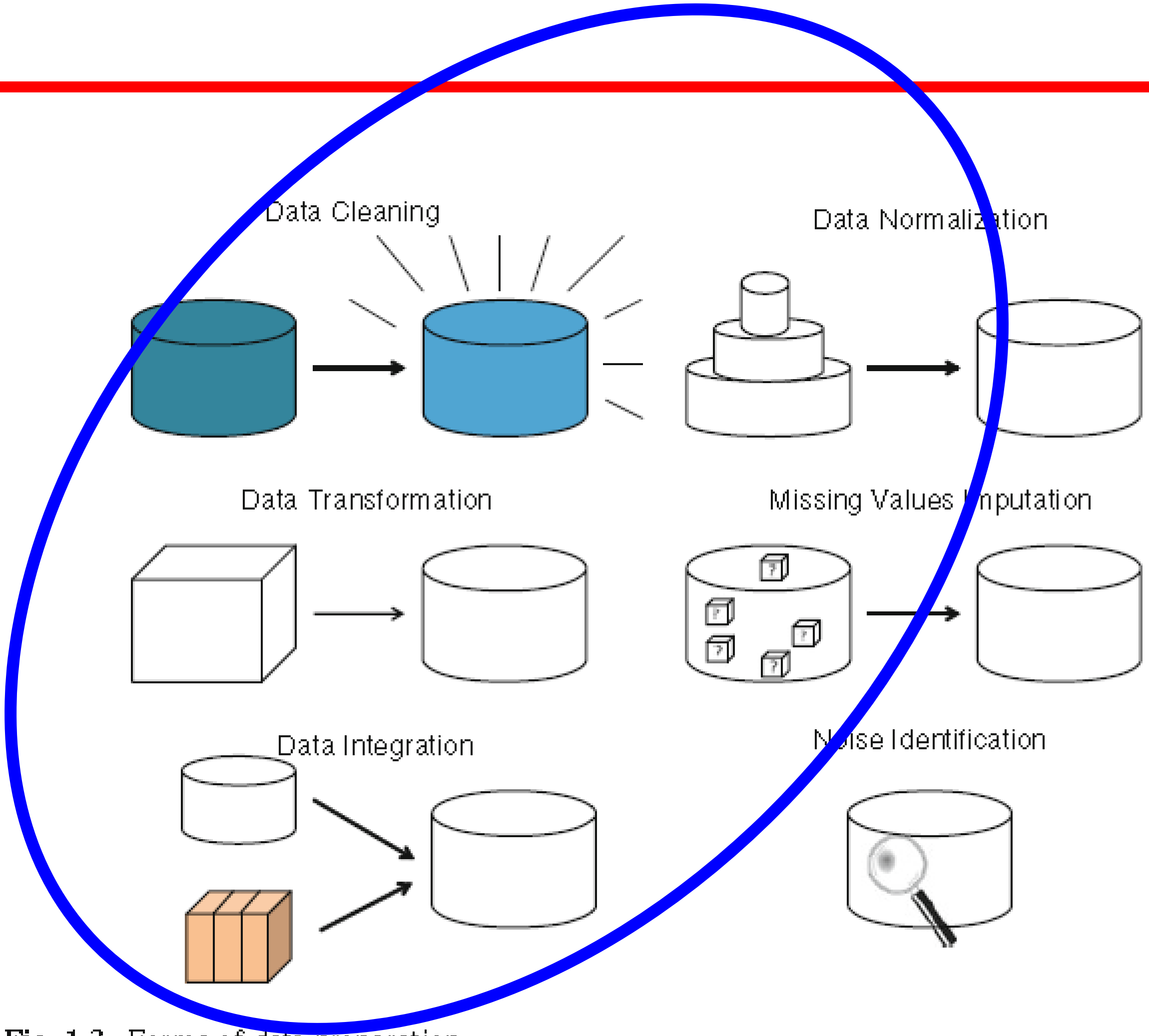
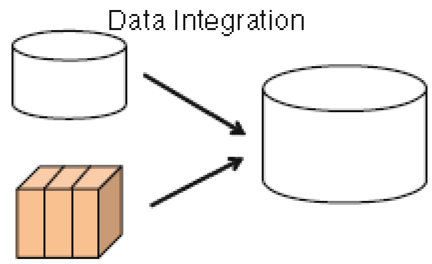
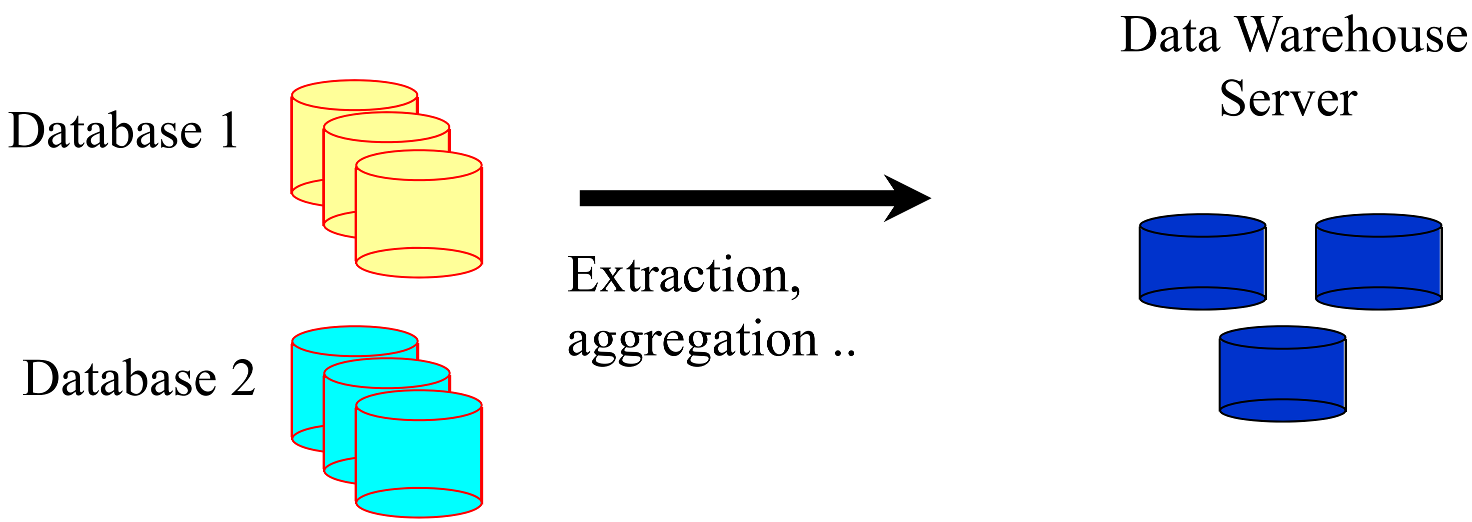


Fig. 1.3 Forms of data preparation

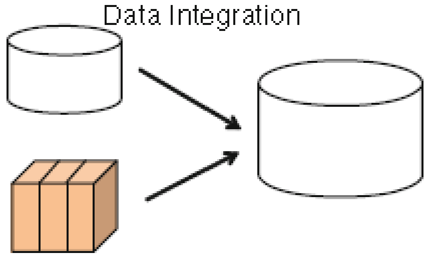
Data Integration



- Obtain data from different information sources.
- Address problems of codification and representation.
- Integrate data from different tables to produce homogeneous information, ...



Data Integration



Examples

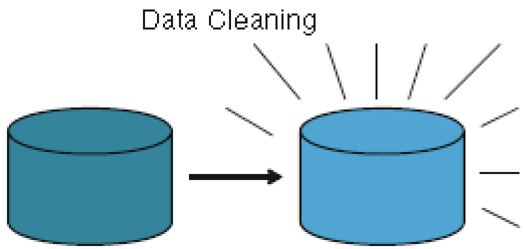
- Different scales: Salary in dollars versus euros (€)

- Derivative attributes: Mensual salary versus annual salary

item	Salary/month
1	5000
2	2400
3	3000

item	Salary
6	50,000
7	100,000
8	40,000

Data Cleaning

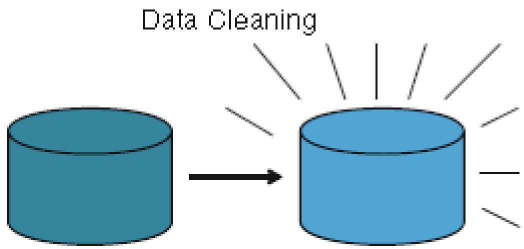


- Objetictives:
  - Fix inconsistencies
  - Fill/impute missing values,
  - Smooth noisy data,
  - Identify or remove *outliers* …

- Some Data Mining algorithms have proper methods to deal with incomplete or noisy data. But in general, these methods are not very robust. It is usual to perform a data cleaning previously to their application.

Bibliography:
W. Kim, B. Choi, E.-D. Hong, S.-K. Kim
A taxonomy of dirty data.
Data Mining and Knowledge Discovery 7, 81-99, 2003.

Data Cleaning



Data cleaning: Example

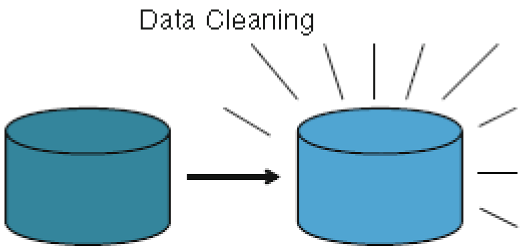
- **Original Data**

**000000000130.06.19971979-10-3080145722 #000310 111000301.01.000100000000004
0000000000000.000000000000000.000000000000000.000000000000000.000000000000000.000000000000000.0000
00000000000. 000000000000000.000000000000000.0000000......
000000000000000.000000000000000.000000000000000.000000000000000.000000000000000.000000000000000.00
0000000000000.000000000000000.000000000000000.000000000000000.000000000000000.000000000000000.0000
00000000000.000000000000000.000000000000000.000000000000000.000000000000000.000000000000000.000000
000000000.000000000000000.000000000000000.000000000000000.00 0000000000300.00 0000000000300.00**

- **Clean Data**

**0000000001,199706,1979.833,8014,5722 , ,#000310 ....
,111,03,000101,0,04,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0300,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0300,0300.00**

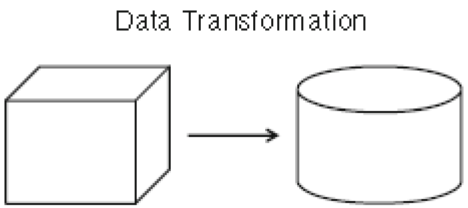
Data Cleaning



Data Cleaning: Inconsistent data

Age="42"
Birth Date="03/07/1997"

Data transformation



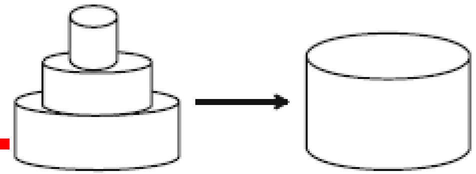
- Objective: To transform data in the best way possible to the application of Data Mining algorithms.

- Some typical operations:
  - Aggregation. i.e. Sum of the totality of month sales in an unique attribute called anual sales,…
  - Data generalization. It is to obtain higher degrees of data from the currently available, by using concept hierarchies.
    - streets → cities
    - Numerical age → {young, adult, half-age, old}
  - Normalization: Change the range [-1,1] or [0,1].
  - Lineal transformations, quadratic, polinominal, …

Bibliography:

T. Y. Lin. Attribute Transformation for Data Mining I: Theoretical Explorations. International Journal of Intelligent Systems 17, 213-222, 2002.

Data Normalization



Normalization

- Objective: convert the values of an attribute to a better range.

- Useful for some techniques such as Neural Networks o distance-based methods (k-Nearest Neighbors,…).

- Some normalization techniques:

  Z-score normalization

$$v' = \frac{v - \overline{A}}{\sigma_A}$$

  min-max normalization: Perform a lineal transformation of the original data.

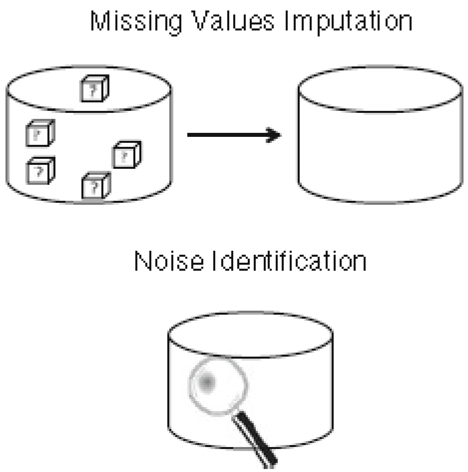
$$[\min_A, \max_A] \rightarrow [new_{\min_A}, new_{\max_A}]$$

$$v' = \frac{v - \min_A}{\max_A - \min_A}(new_{\max_A} - new_{\min_A}) + new_{\min_A}$$

  The relationships among original data are maintained.

Data Preprocessing in Data Mining

1. Introduction. Data Preprocessing
2. Integration, Cleaning and Transformations
3. Imperfect Data
4. Data Reduction
5. Final Remarks



Imperfect data

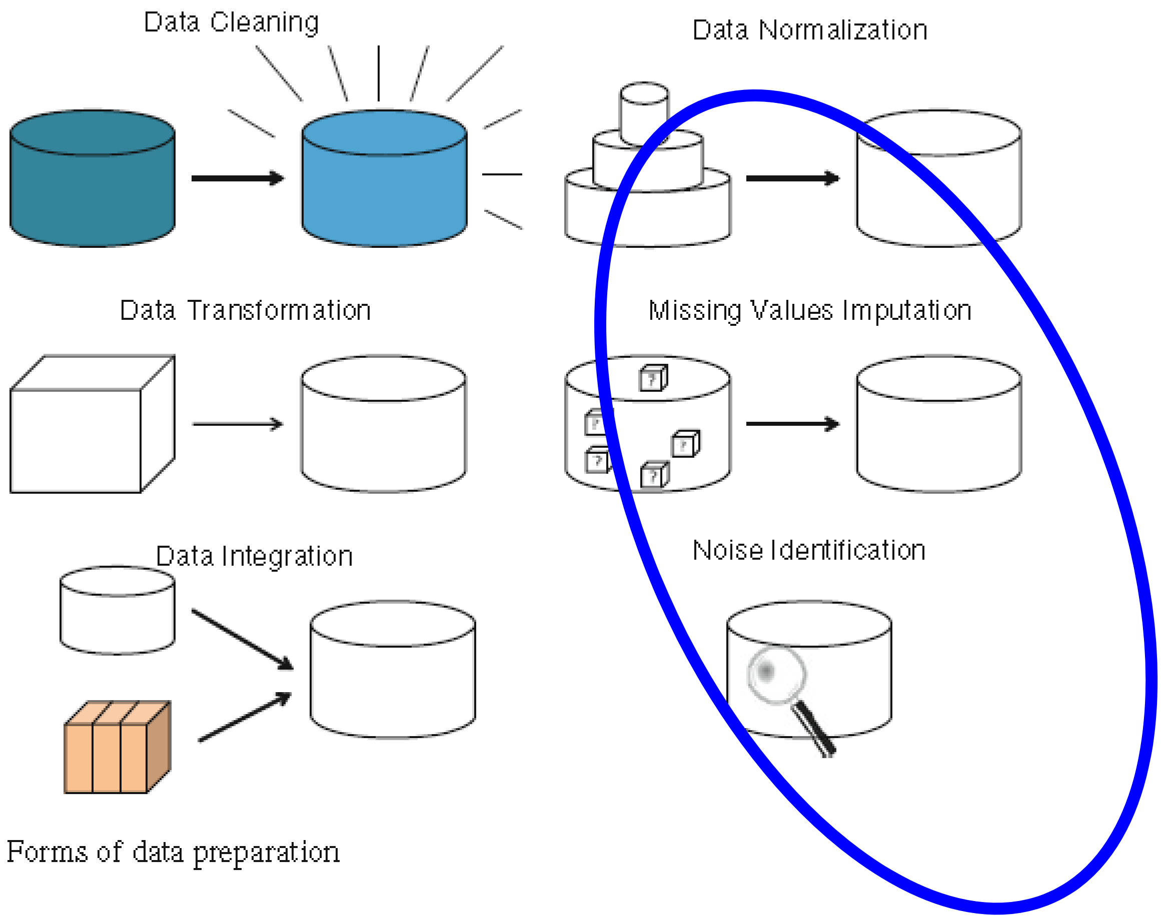
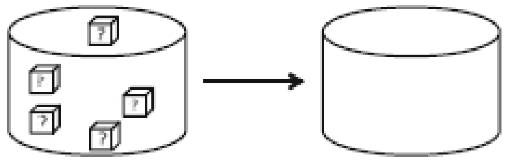


Fig. 1.3 Forms of data preparation

Missing values

Missing Values Imputation

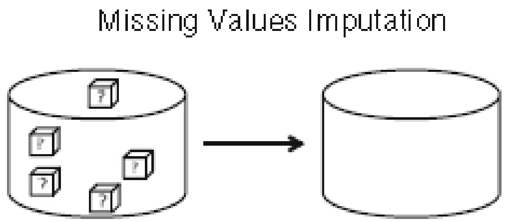


Attributes

Instances

	1	2	3	4	5	…	m
1					?		
2			?				
3		?		?			
4							
5							
6						?	
7			?		?		
8							
9							
10			?			?	
11		?					
⋮				?			
n							?

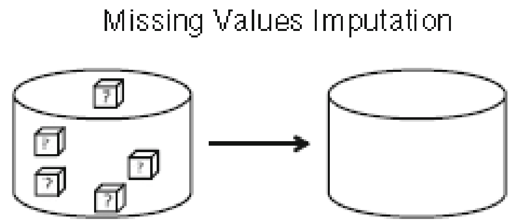
Missing values



It could be used the next choices, although some of them may skew the data:

- Ignore the tuple. It is usually used when the variable to classify has no value.
- Use a global constant for the replacement. I.e. “unknown”,”?”,…
- Fill tuples by means of mean/deviation of the rest of the tuples.
- Fill tuples by means of mean/deviation of the rest of the tuples belonging to the same class.
- Impute with the most probable value. For this, some technique of inference could be used, i.e., bayesian or decision trees.

Missing values



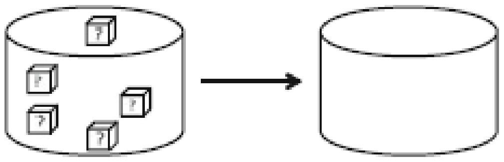
MISSING VALUES

Full Name	Short Name	Reference
Delete Instances with Missing Values	Ignore-MV	P.A. Gourraud, E. Ginin, A. Cambon-Thomsen. Handling Missing Values In Population Data: Consequences For Maximum Likelihood Estimation Of Haplotype Frequencies. European Journal of Human Genetics 12:10 (2004) 805-812.
Event Covering Synthesizing	EventCovering-MV	D.K.Y. Chiu, A.K.C. Wong. Synthesizing Knowledge: A Cluster Analysis Approach Using Event-Covering. IEEE Transactions on Systems, Man and Cybernetics, Part B 16:2 (1986) 251-259.
K-Nearest Neighbor Imputation	KNN-MV	G.E.A.P.A. Batista, M.C. Monard. An Analysis Of Four Missing Data Treatment Methods For Supervised learning. Applied Artificial Intelligence 17:5 (2003) 519-533.
Most Common Attribute Value	MostCommon-MV	J.W. Grzymala-Busse, L.K. Goodwin, W.J. Grzymala-Busse, X. Zheng. Handling Missing Attribute Values in Preterm Birth Data Sets. 10th International Conference of Rough Sets, Fuzzy Sets, Data Mining and Granular Computing (RSFDGrC'05). LNCS 3642, Springer 2005, Regina (Canada, 2005) 342-351.
Assign All Posible Values of the Attribute	AllPossible-MV	J.W. Grzymala-Busse. On the Unknown Attribute Values In Learning From Examples. 6th International Symposium on Methodologies For Intelligent Systems (ISMIS91). Charlotte (USA, 1991) 368-377.
K-means Imputation	KMeans-MV	J. Deogun, W. Spaulding, B. Shuart, D. Li. Towards Missing Data Imputation: A Study of Fuzzy K-means Clustering Method. 4th International Conference of Rough Sets and Current Trends in Computing (RSCTC'04). LNCS 3066, Springer 2004, Uppsala (Sweden, 2004) 573-579.
Concept Most Common Attribute Value	ConceptMostCommon-MV	J.W. Grzymala-Busse, L.K. Goodwin, W.J. Grzymala-Busse, X. Zheng. Handling Missing Attribute Values in Preterm Birth Data Sets. 10th International Conference of Rough Sets, Fuzzy Sets, Data Mining and Granular Computing (RSFDGrC'05). LNCS 3642, Springer 2005, Regina (Canada, 2005) 342-351.

15 methods

http://www.keel.es/

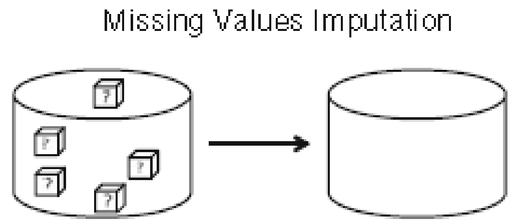
Missing values



Algorithm 3 kNNI algorithm.

function kNNI(T - dataset with MVs, k - number of neighbors per instance to be chosen, $D(x, y)$ - a distance or dissimilarity function of x and y, S - the imputed version of T)
  initialize: $S = \{\}$
  for each instance y_i in T **do**
    $\widehat{y}_i \leftarrow y_i$
    if y_i contains any missing value **then**
      Find set I_{Ki} with the k nearest instances to y_i from T using D
      for each missing value in attribute h of y_i **do**
        if h is numerical **then**
          $\widehat{y}_{ih} = \left(\sum_{j \in I_{Kih}} y_{jh}\right) / (|I_{Kih}|)$
        else
          $\widehat{y}_{ih} = mode(I_{Kih})$
        end if
      end for
    end if
    $S \leftarrow \widehat{y}_{ih}$
  end for
  return S
end function

Missing values



Bibliography:

WEBSITE: http://sci2s.ugr.es/MVDM/

J. Luengo, S. García, F. Herrera, **A Study on the Use of Imputation Methods for Experimentation with Radial Basis Function Network Classifiers Handling Missing Attribute Values: The good synergy between RBFs and EventCovering method**. *Neural Networks, doi:10.1016/j.neunet.2009.11.014, 23(3) (2010) 406-418*.

S. García, F. Herrera, **On the choice of the best imputation methods for missing values considering three groups of classification methods**. *Knowledge and Information Systems 32:1 (2012) 77-108, doi:10.1007/s10115-011-0424-2*

Noise cleaning

Types of examples

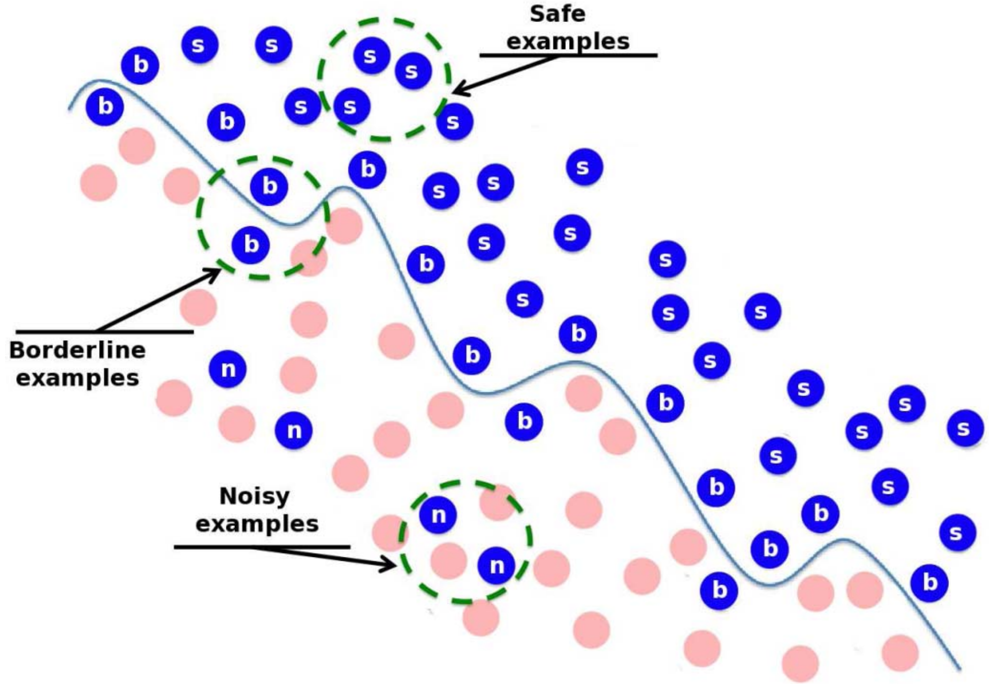
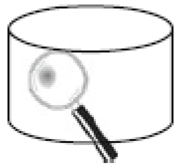


Fig. 5.2 The three types of examples considered in this book: safe examples (labeled as *s), borderline* examples (labeled as *b) and noisy examples (labeled as n). The continuous line shows the* decision boundary between the two classes



Noise cleaning

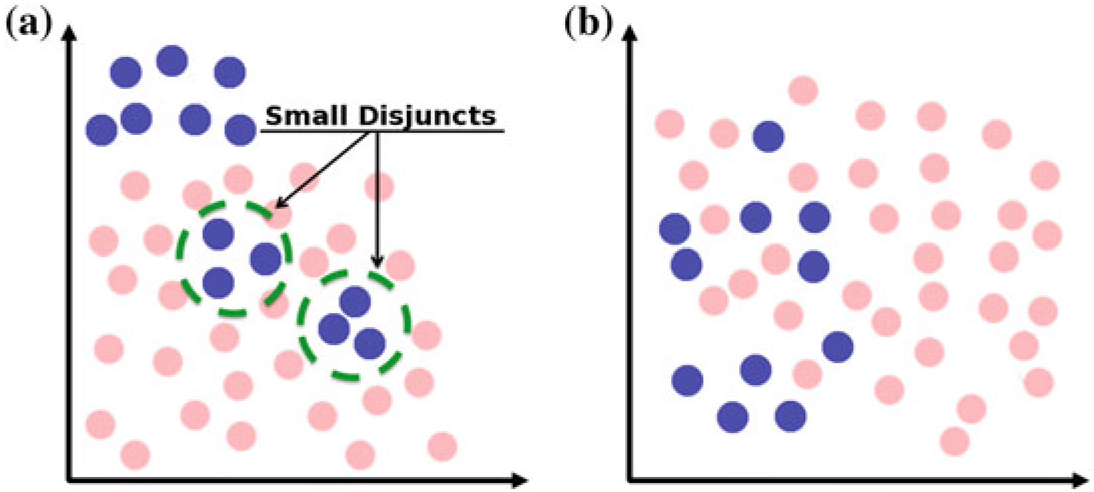
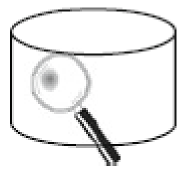


Fig. 5.1 Examples of the interaction between classes: a) small disjuncts and b) overlapping between classes

Noise Identification



Noise cleaning

Use of noise filtering techniques in classification

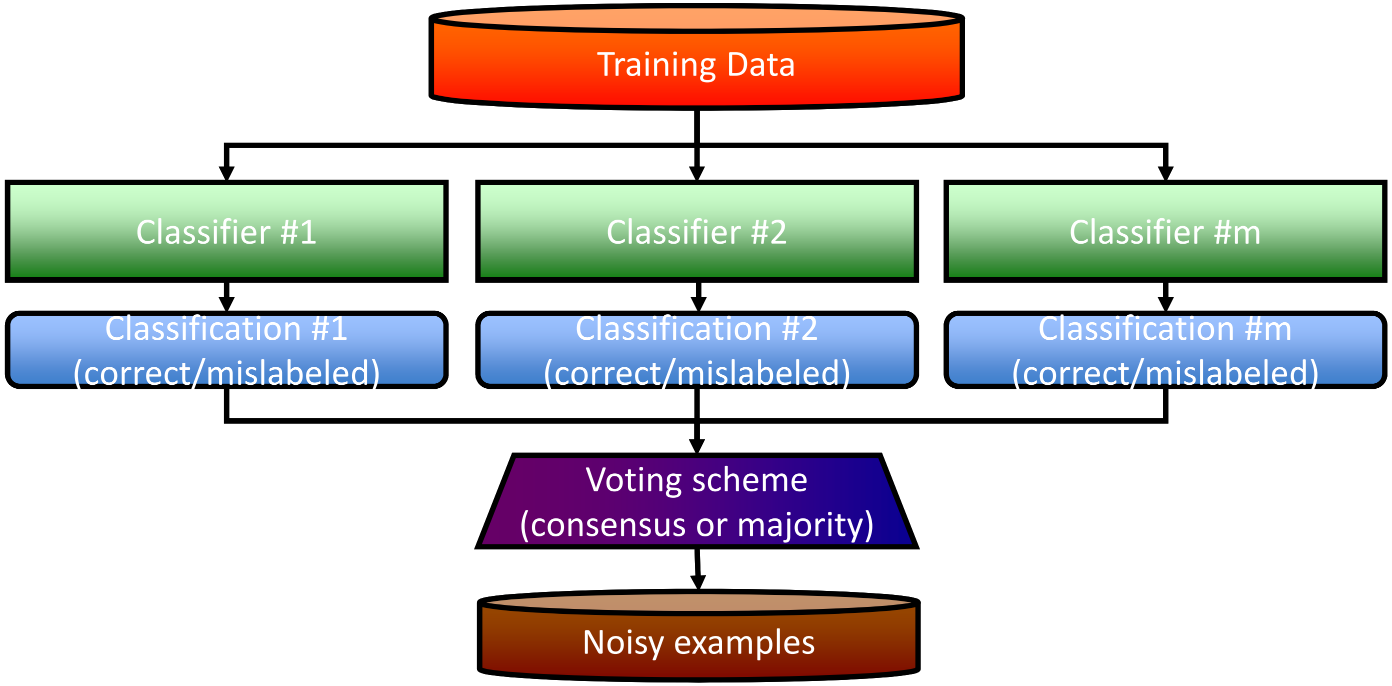
The three noise filters mentioned next, which are the most-known, use a voting scheme to determine what cases have to be removed from the training set:

- ***Ensemble Filter (EF)***
- ***Cross-Validated Committees Filter***
- ***Iterative-Partitioning Filter***

Ensemble Filter (EF)

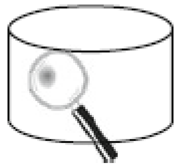
- C.E. Brodley, M.A. Friedl. **Identifying Mislabeled Training Data**. *Journal of Artificial Intelligence Research 11 (1999) 131-167*.
- **Different learning algorithm** (C4.5, 1-NN and LDA) are used to create classifiers in several subsets of the training data that serve as noise filters for the training sets.
- Two main steps:

1. For each learning algorithm, a ***k-fold cross-validation*** is used to tag each training example as **correct** (prediction = training data label) or **mislabeled** (prediction ≠ training data label).
2. A ***voting scheme*** is used to identify the final set of noisy examples.
    - **Consensus voting**: it removes an example if it is misclassified by all the classifiers.
    - **Majority voting**: it removes an instance if it is misclassified by more than half of the classifiers.



Ensemble Filter (EF)

Noise Identification



Algorithm 4 EF algorithm.

function EF(T - dataset with MVs, Γ - number of subsets, μ - number of filters to be used, F - set of classifiers)

Split the training data set T into $T_i, i = 1 \ldots \Gamma$ equal sized subsets

for each filter $F_x, x = 1 to \mu$ **do**

for each subset T_i **do**

Use $\{T_j, j \neq i\}$ to train F_x resulting in F_x^i

for each instance t in T_i **do**

Classify t with every F_x^i

end for

end for

end for

for each instance t in T **do**

Use a voting scheme to include t in T_N according to the classifications made by each filter F_x

end for

return $T - T_N$

end function

Cross-Validated Committees Filter (CVCF)

- S. Verbaeten, A.V. Assche. **Ensemble methods for noise elimination in classification problems**. *4th International Workshop on Multiple Classifier Systems (MCS 2003). LNCS 2709, Springer 2003, Guilford (UK, 2003) 317-325*.

- CVCF is similar to EF ➔ two main differences:

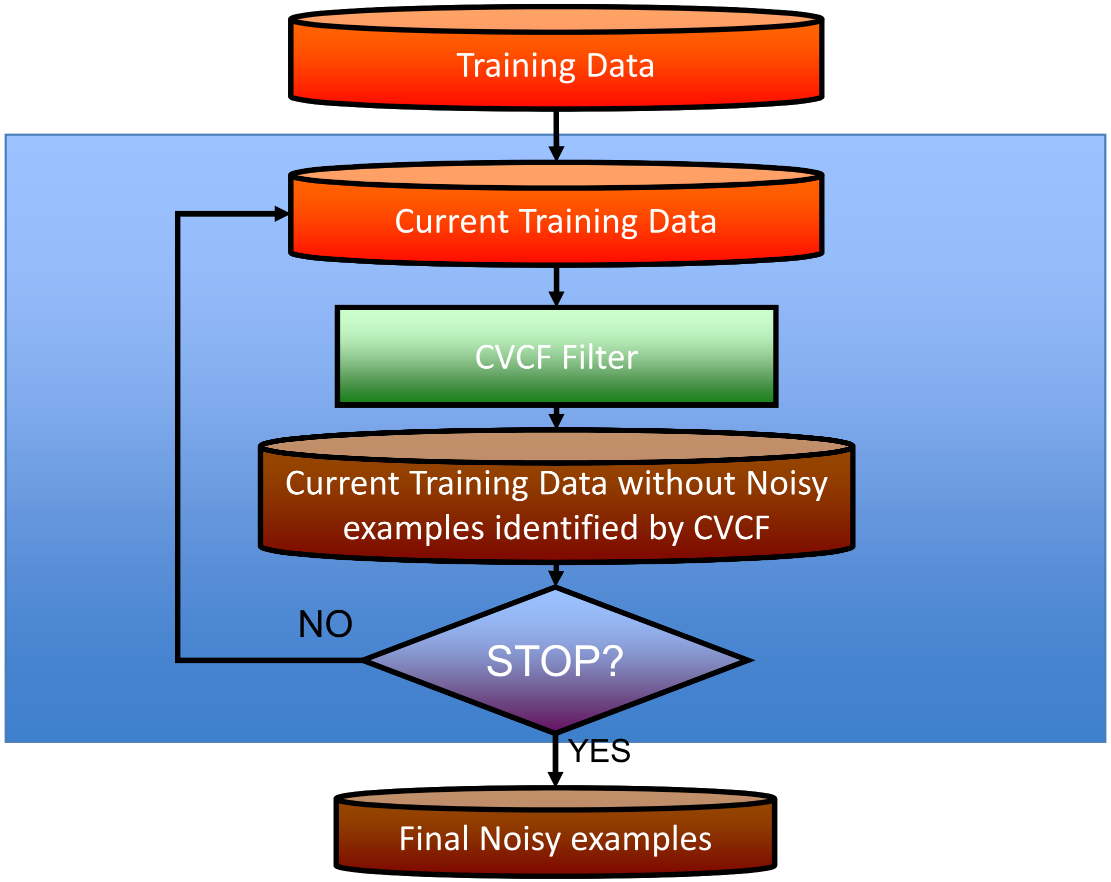
1. **The same learning algorithm (C4.5)** is used to create classifiers in several subsets of the training data.

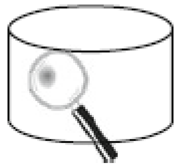
   The authors of CVCF place special emphasis on using **ensembles of decision trees** such as C4.5 because they work well as a filter for noisy data.

2. Each classifier built with the ***k-fold cross-validation*** is used to tag **ALL the training examples** (not only the test set) as **correct** (prediction = training data label) or **mislabeled** (prediction ≠ training data label).

Iterative Partitioning Filter (IPF)

- T.M. Khoshgoftaar, P. Rebours. **Improving software quality prediction by noise filtering techniques**. *Journal of Computer Science and Technology 22 (2007) 387-396.*
- IPF removes noisy data in **multiple iterations** using **CVCF** until a stopping criterion is reached.
- The iterative process stops if, for a number of consecutive iterations, the number of noisy examples in each iteration is less than a percentage of the size of the training dataset.





Noise cleaning

NOISY DATA FILTERING

Full Name	Short Name	Reference
Saturation Filter	SaturationFilter-F	D. Gamberger, N. Lavrac, S. Dzroski. Noise detection and elimination in data preprocessing: Experiments in medical domains. Applied Artificial Intelligence 14:2 (2000) 205-223.
Pairwise Attribute Noise Detection Algorithm Filter	PANDA-F	J.D. Hulse, T.M. Khoshgoftaar, H. Huang. The pairwise attribute noise detection algorithm. Knowledge and Information Systems 11:2 (2007) 171-190.
Classification Filter	ClassificationFilter-F	D. Gamberger, N. Lavrac, C. Groselj. Experiments with noise filtering in a medical domain. 16th International Conference on Machine Learning (ICML99). San Francisco (USA, 1999) 143-151.
Automatic Noise Remover	ANR-F	X. Zeng, T. Martinez. A Noise Filtering Method Using Neural Networks. IEEE International Workshop on Soft Computing Techniques in Instrumentation, Measurement and Related Applications (SCIMA2003). Utah (USA, 2003) 26-31.
Ensemble Filter	EnsembleFilter-F	C.E. Brodley, M.A. Friedl. Identifying Mislabeled Training Data. Journal of Artificial Intelligence Research 11 (1999) 131-167.
Cross-Validated Committees Filter	CVCommitteesFilter-F	S. Verbaeten, A.V. Assche. Ensemble methods for noise elimination in classification problems. 4th International Workshop on Multiple Classifier Systems (MCS 2003). LNCS 2709, Springer 2003, Guilford (UK, 2003) 317-325.
Iterative-Partitioning Filter	IterativePartitioningFilter-F	T.M. Khoshgoftaar, P. Rebours. Improving software quality prediction by noise filtering techniques. Journal of Computer Science and Technology 22 (2007) 387-396.

http://www.keel.es/

Data Preprocessing in Data Mining

1. Introduction. Data Preprocessing
2. Integration, Cleaning and Transformations
3. Imperfect Data
4. Data Reduction
5. Final Remarks

Data Reduction

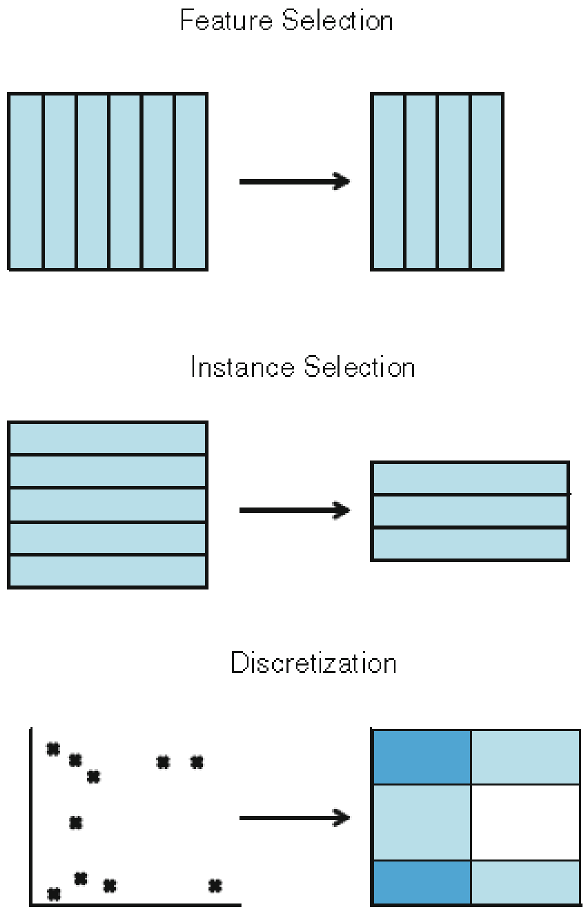
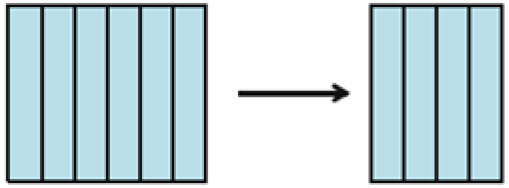


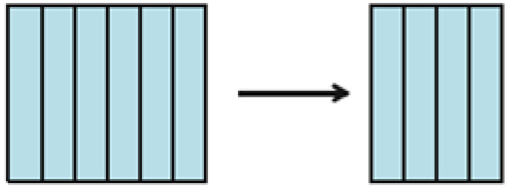
Fig. 1.4 Forms of data reduction

Feature Selection



The problem of *Feature Subset Selection (FSS*) consists of finding a subset of the attributes/features/variables of the data set that optimizes the probability of success in the subsequent data mining taks.

Feature Selection



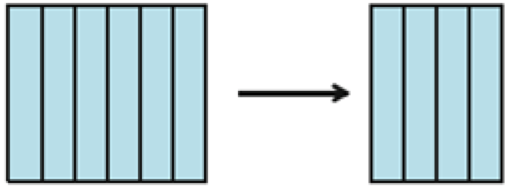
Var. 1.

Var. 5

Var. 13

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
A	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
B	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
C	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
D	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
E	0	1	0	0	0	1	1	0	1	1	0	0	0	0	1	0
F	1	1	1	0	1	1	0	0	1	0	1	0	0	1	0	0

Feature Selection

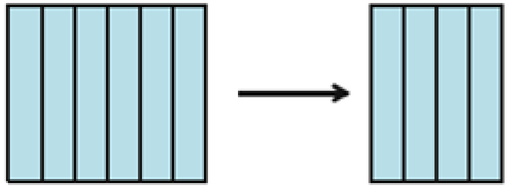


The problem of *Feature Subset Selection (FSS)* consists of finding a subset of the attributes/features/variables of the data set that optimizes the probability of success in the subsequent data mining taks.

Why is feature selection necessary?

- More attributes do not mean more success in the data mining process.
- Working with less attributes reduces the complexity of the problem and the running time.
- With less attributes, the generalization capability increases.
- The values for certain attributes may be difficult and costly to obtain.

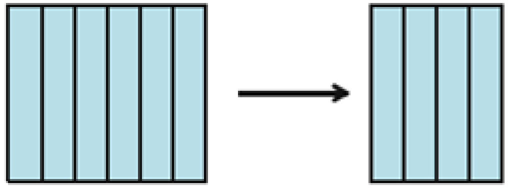
Feature Selection



- The outcome of FS would be:
  - Less data → algorithms couls learn quickly
  - Higher accuracy → the algorithm better generalizes
  - Simpler results → easier to understand them

- **FS has as extension the extraction and construction of attributes.**

Feature Selection



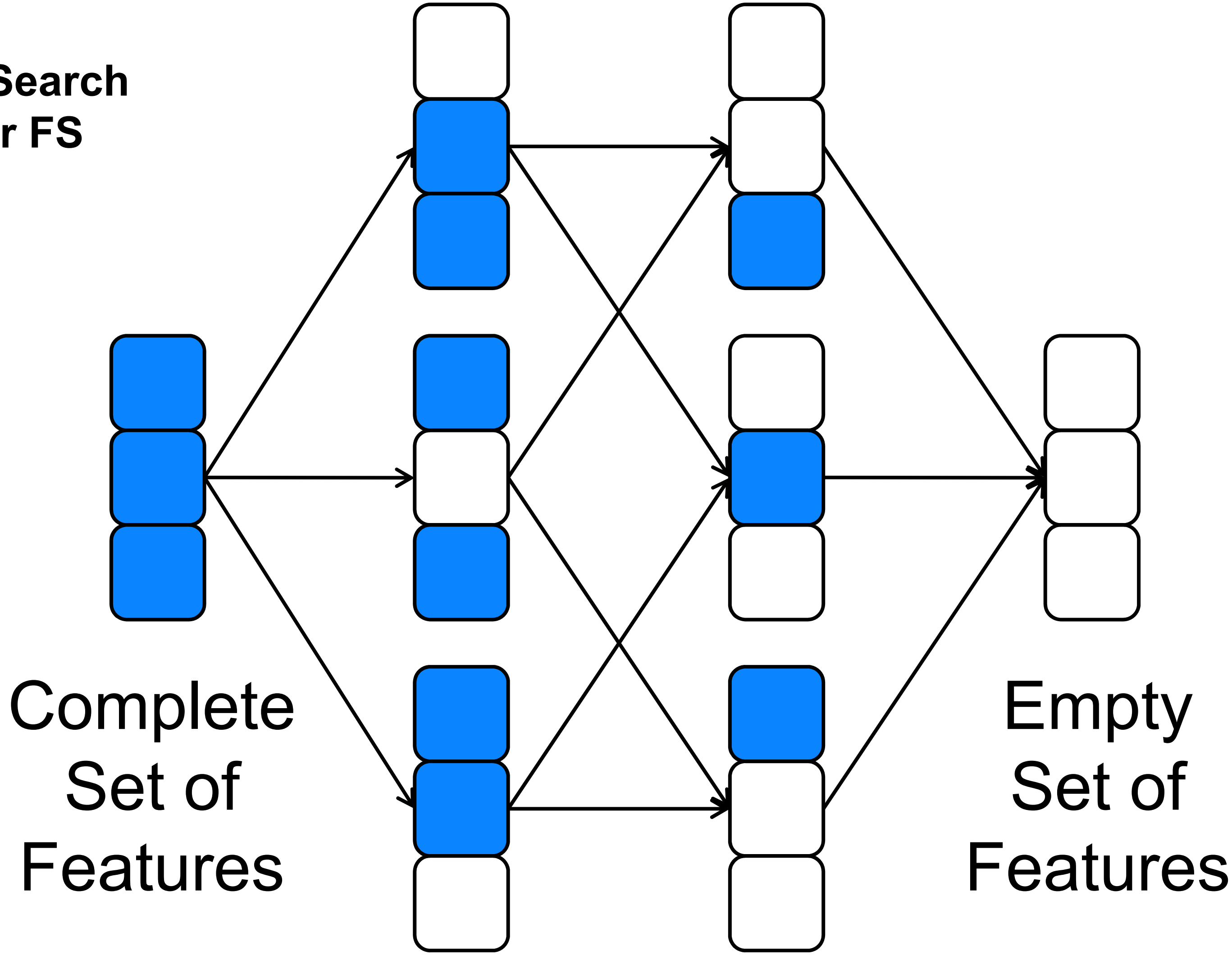
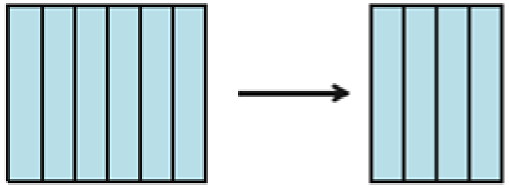
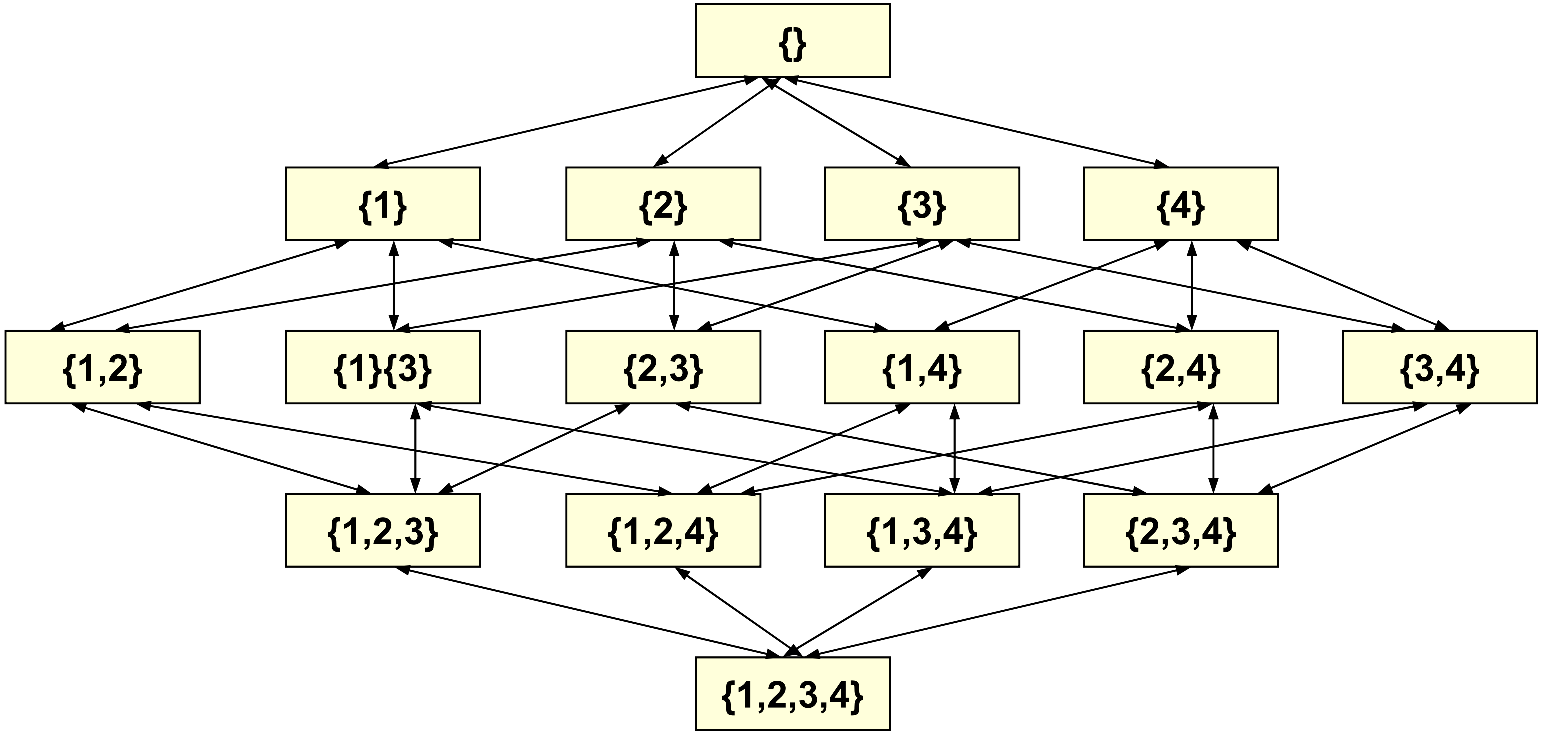


Fig. 7.1 Search space for FS

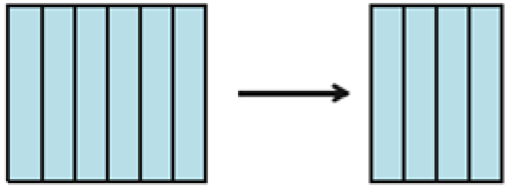
Feature Selection



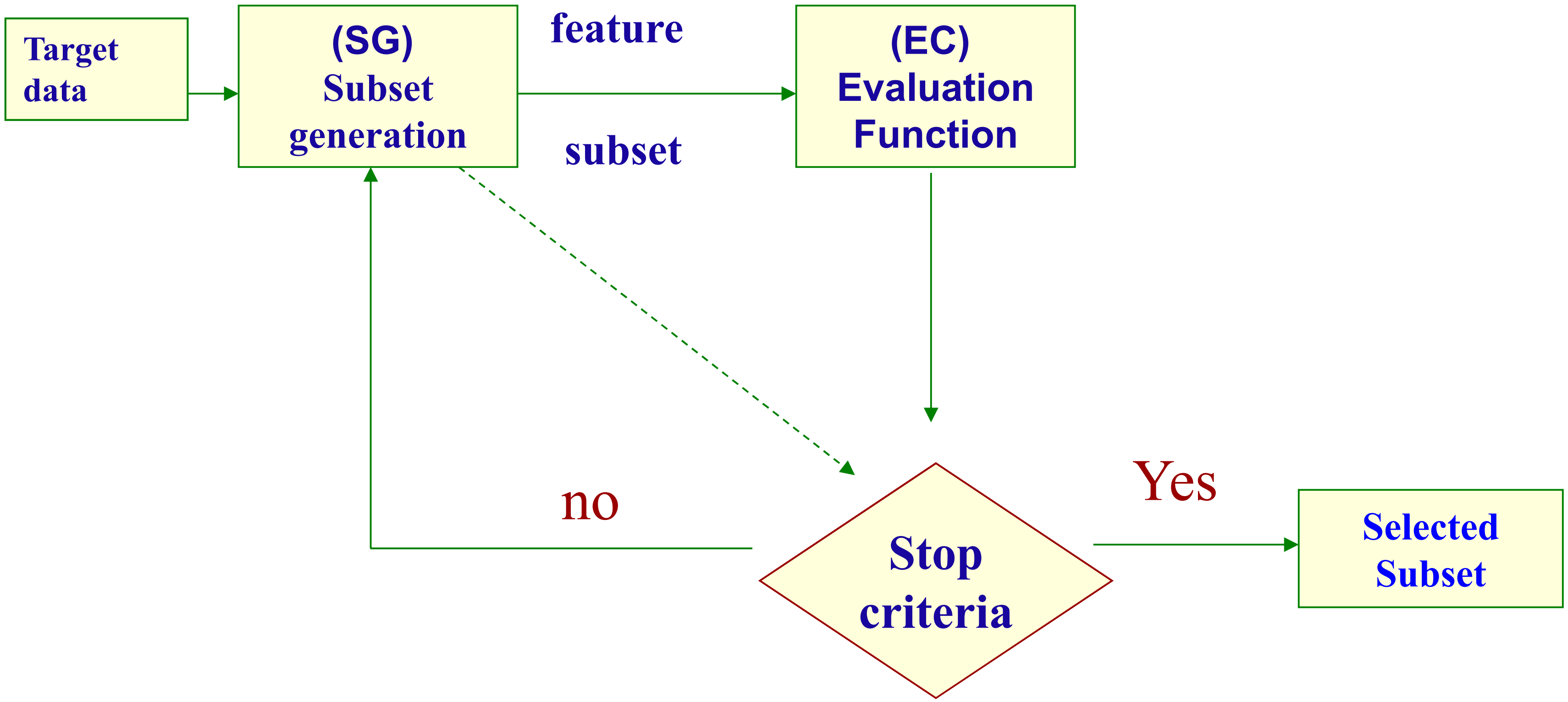
It can be considered as a search problem



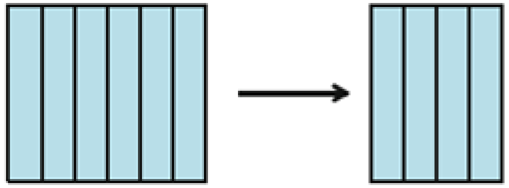
Feature Selection



Process



Feature Selection

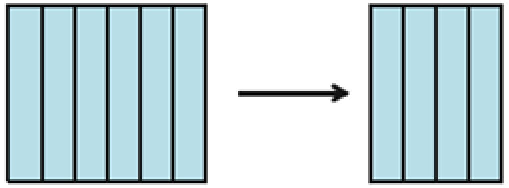


Goal functions: There are two different approaches

- **Filter.** The goal function evaluates the subsets basing on the information they contain. Measures of class separability, statistical dependences, information theory,… are used as the goal function.

- **Wrapper.** The goal function consists of applying the same learning technique that will be used later over the data resulted from the selection of the features. The returned value usually is the accuracy rate of the constructed classifier.

Feature Selection



Process

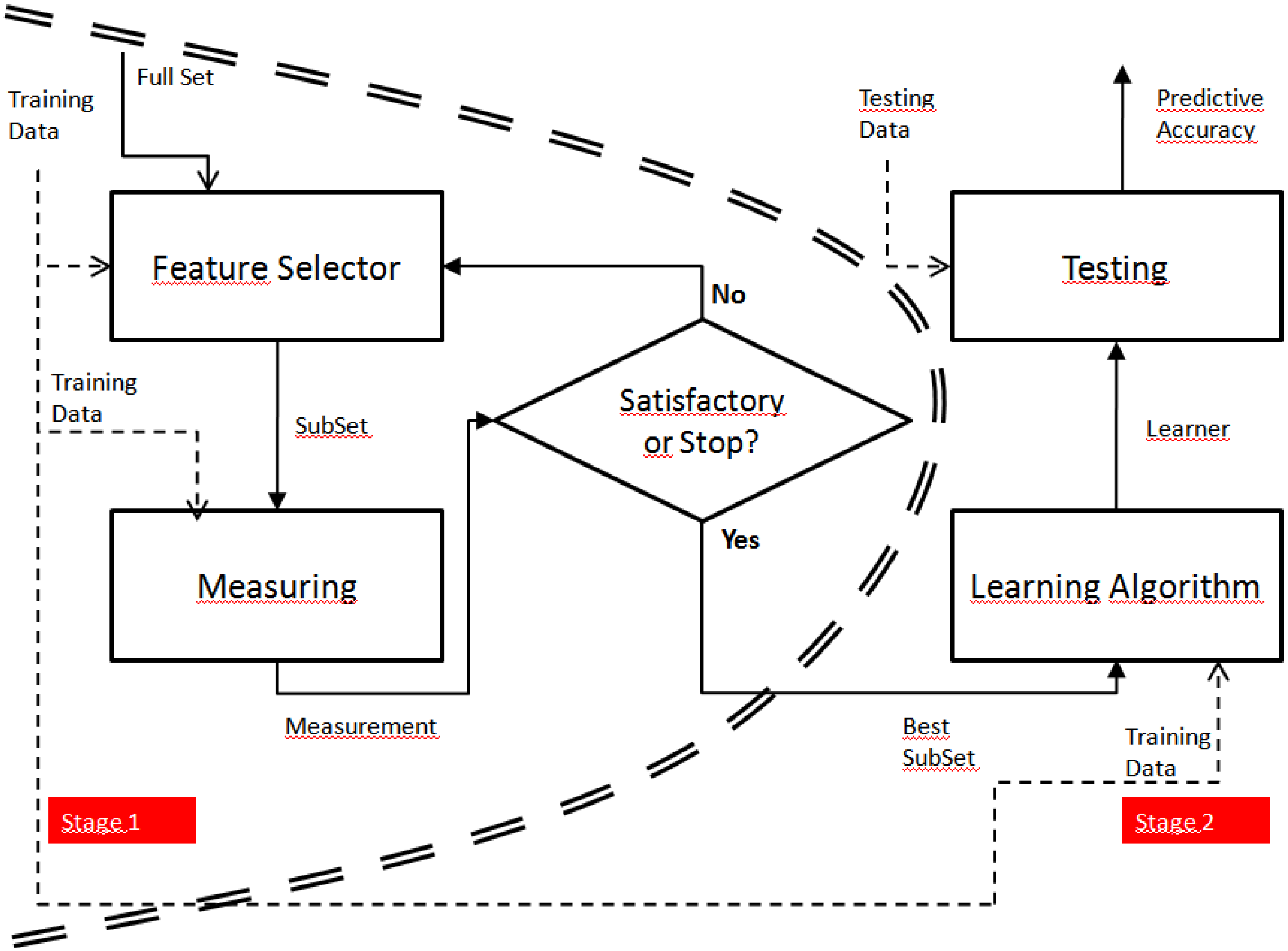
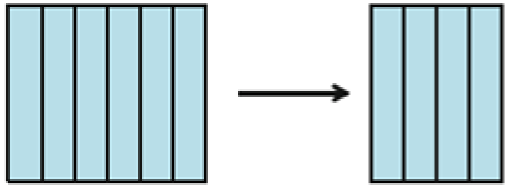


Fig. 7.2 A filter model for FS

Feature Selection



Filtering measures

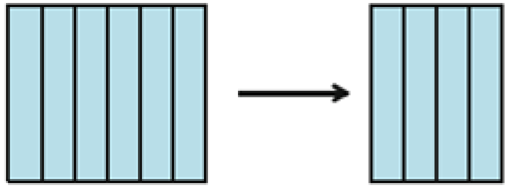
- **Separability measures.** They estimate the separability among classes: euclidean, Mahalanobis,…
  - I.e. In a two-class problem, a FS process based on this kind of measures determined that X is bettern than Y if X induces a greater difference than Y between the two prior conditional probabilities between the classes.

- **Correlation.** Good subset will be those correlated with the class variable

$$f(X_1,\ldots,X_M)=\frac{\sum_{i=1}^{M}\rho_{ic}}{\sum_{i=1}^{M}\sum_{j=i+1}^{M}\rho_{ij}}$$

where ρ_{ic} is the coefficient of correlation between the variable X_i and the label c of the class (C) and ρ_{ij} is the correlation coefficient between X_i and X_j

Feature Selection



- **Information theory based measures**
  - Correlation only can estimate lineal dependences. A more powerful method is the mutual information $I(X_{1,\ldots,M}; C)$

$$f(X_{1,\ldots,M}) = I(X_{1,\ldots,M}; C) = H(C) - H(C \mid X_{1,\ldots,M}) =$$
$$\sum_{c=1}^{|C|} \int_{X_{1,\ldots,M}} P(X_{1\ldots M}, \omega_c) \log \frac{P(X_{1\ldots M}, \omega_c)}{P(X_{1\ldots M}) P(\omega_c)} dx$$

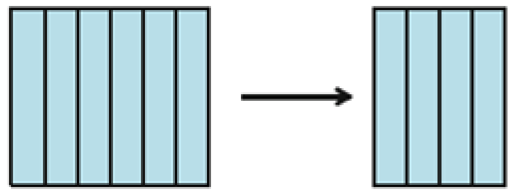
where H represents the entropy and ω_c the c-th label of the class C

  - Mutual information measures the quantity of uncertainty that decreases in the class C when the values of the vector $X_{1\ldots M}$ are known.
  - Due to the complexity of the computation of I, it is usual to use heurisctics rules

$$f(X_{1\ldots M}) = \sum_{i=1}^{M} I(X_i; C) - \beta \sum_{i=1}^{M} \sum_{j=i+1}^{M} I(X_i; X_j)$$

with β=0.5, as example.

Feature Selection

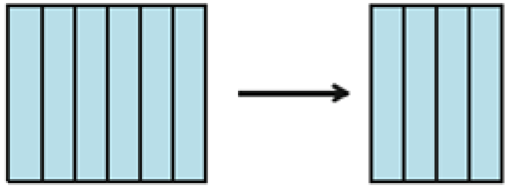


- **Consistency measures**

  - The three previous groups of measures try to find those features than could, maximally, predict the class better than the remain.
    - This approach cannot distinguish between two attributes that are equally appropriate, it does not detect redundant features.

  - Consistency measures try to find a minimum number of features that are able to separate the classes in the same way that the original data set does.

Feature Selection



Process

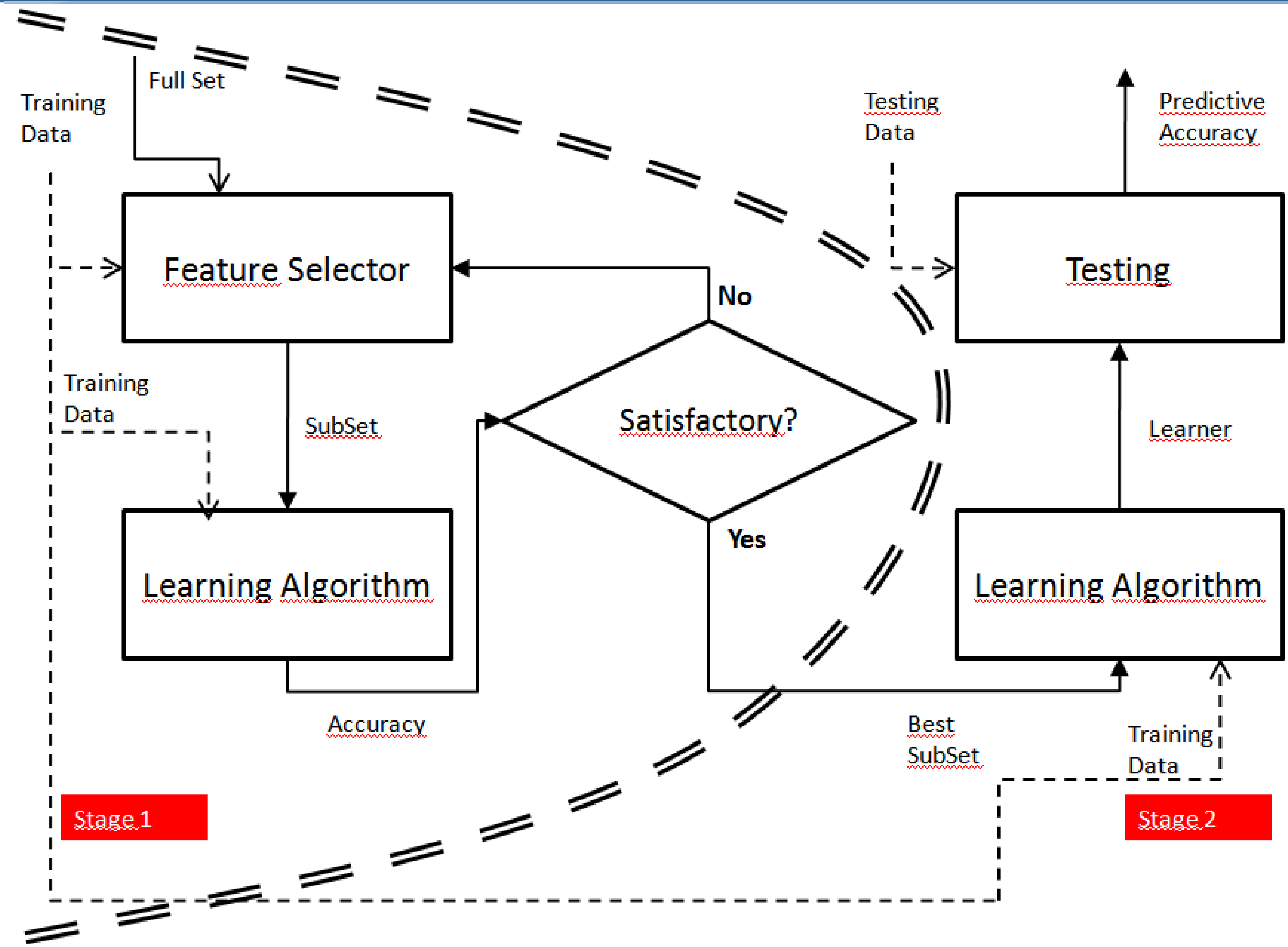
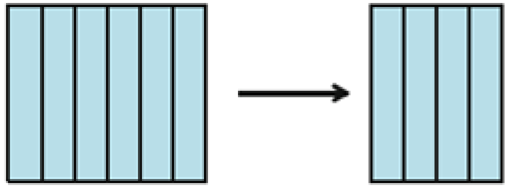


Fig. 7.2 A wrapper model for FS

Feature Selection



Process

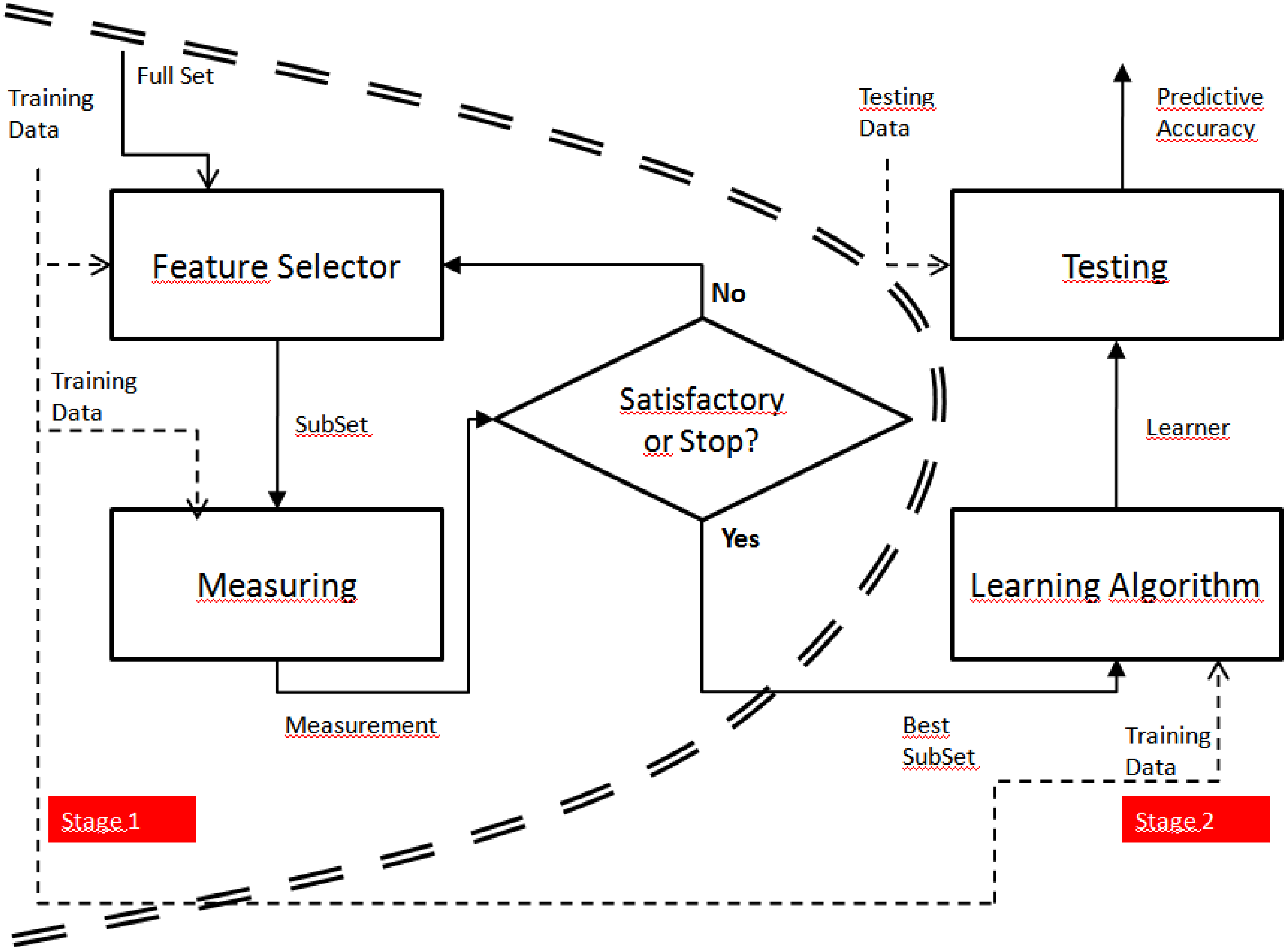
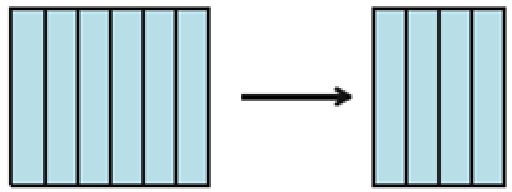


Fig. 7.2 A filter model for FS

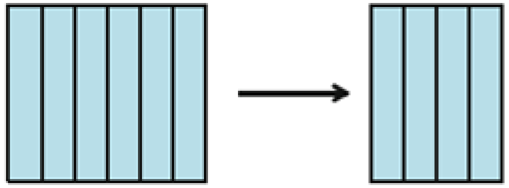
Feature Selection



Advantages

- **Wrappers:**
  - **Accuracy:** generally, they are more accurate than filters, due to the intercation between the classifier used in the goal function and the training data set.
  - **Generalization capability:** they pose capacity to avoid overfitting due to validation techniques employed.
- **Filters:**
  - **Fast:** They usually compute frequencies, much quicker than training a classifier.
  - **Generality:** Due to they evaluate instrinsic properties of the data and not their interaction with a classifier, they can be used in any problem.

Feature Selection

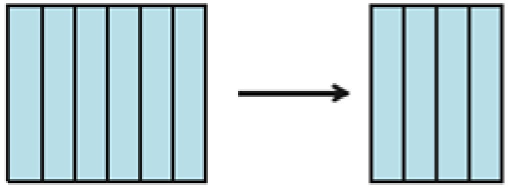


Drawbacks

- **Wrappers:**
  - **Very costly:** for each evaluation, it is required to learn and validate a model. It is prohibitive to complex classifiers.
  - **Ad-hoc solutions:** The solutions are skewed towards the used classifier.

- **Filters:**
  - **Trend to include many variables:** Normally, it is due to the fact that there are monotone features in the goal function used.
    - The use should set the threshold to stop.

Feature Selection



Categories

1. According to evaluation:

   filter

   wrapper

2. Class availability:

   Supervised

   Unsupervised

3. According to the search:

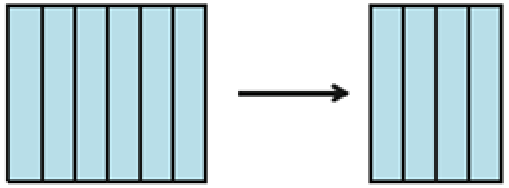
   Complete $O(2^N)$
   Heurístic $O(N^2)$
   Random ??

4. According to outcome:

   Ranking

   Subset of features

Feature Selection



Algorithms for getting subset of features

They returns a subset of attributes optimized according to an evaluation criterion.

Input: x attributes – U evaluation criterion

Subset = {}

Repeat

    S_k = generateSubset(x)

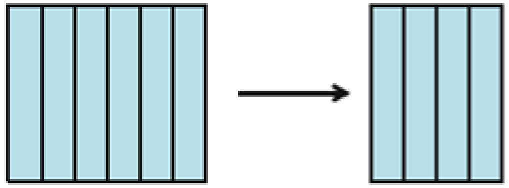
    if improvement(S, S_k,U)

        Subset = S_k

Until StopCriterion()

Output: List, of the most relevant atts.

Feature Selection



Ranking algorithms

They return a list of attributes sorted by an evaluation criterion.

Input: x attributed – U evaluation criterion

List = {}

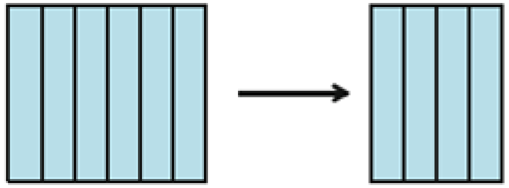
For each Attribute x_i, $i \in \{1,...,N\}$

v_i = compute(x_i,U)

set x_i within the List according to v_i

Output: List, more relevant atts first

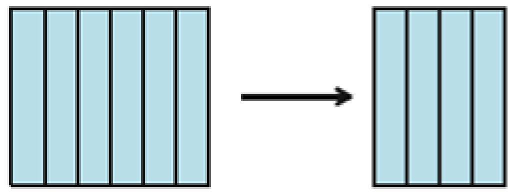
Feature Selection



Ranking algorithms

Attributes	A1	A2	A3	A4	A5	A6	A7	A8	A9
Ranking	A5	A7	A4	A3	A1	A8	A6	A2	A9
	A5	A7	A4	A3	A1	A8	(6 attributes)		

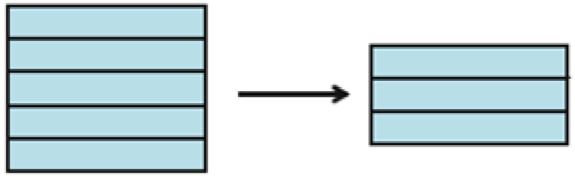
Feature Selection



Some relevant algorithms:

- Focus algorithm. Consistency measure for forward search
- Mutual Information based Features Selection (MIFS).
- mRMR: Minimum Redundancy Maximum Relevance
- Las Vegas Filter (LVF)
- Las Vegas Wrapper (LVW)
- Relief Algorithm

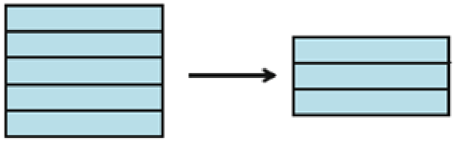
Instance Selection



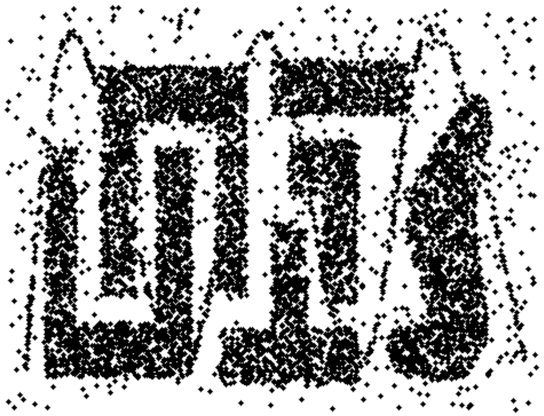
- Instance selection try to choose the examples which are relevant to an application, achieving the maximum performance. The outcome of IS would be:
  - Less data → algorithms learn quicker
  - Higher accuracy → the algorithm better generalizes
  - Simpler results → easier to understand them

- **IS has as extension the generation of instances (prototype generation)**

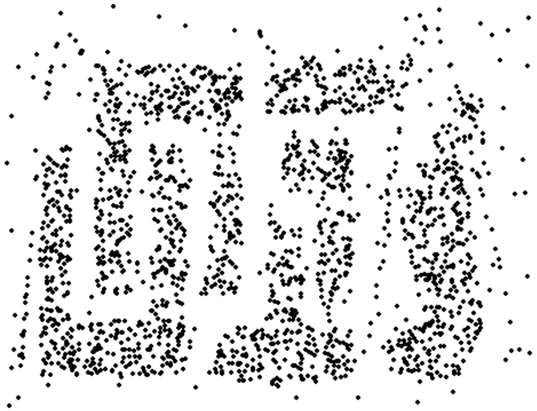
Instance Selection



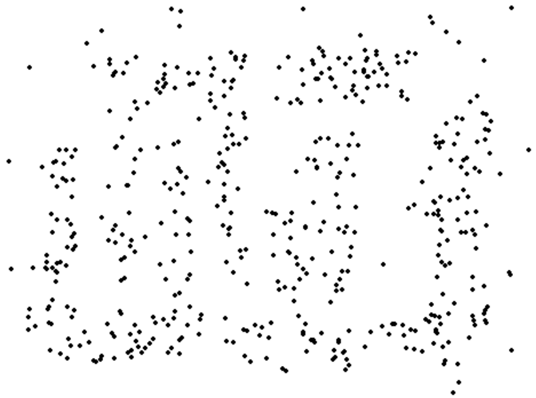
Different size examples



8000 points

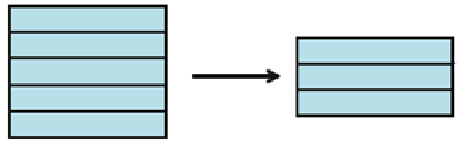


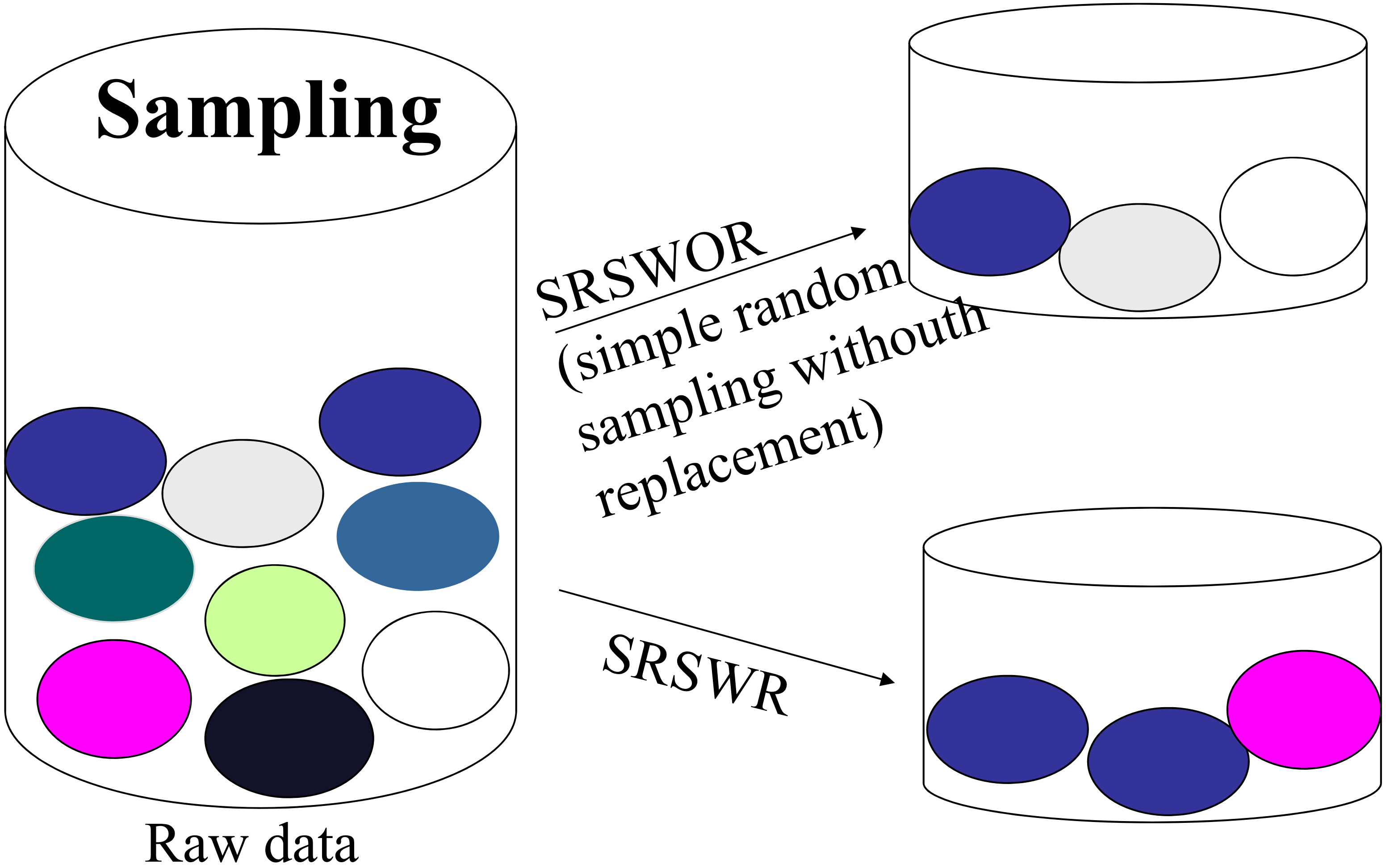
2000 points



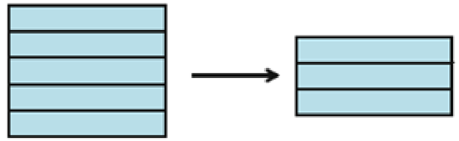
500 points

Instance Selection



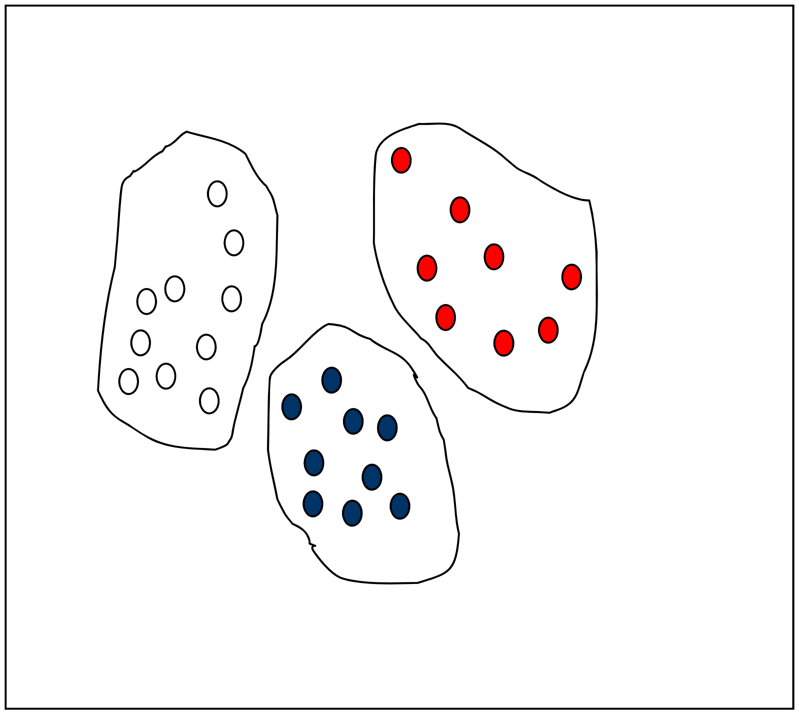


Instance Selection

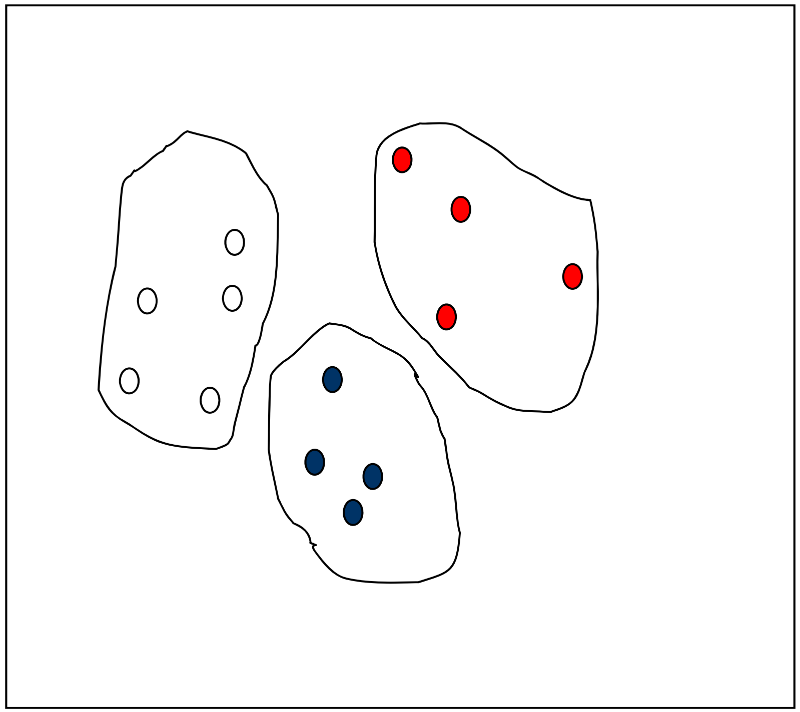


Sampling

Raw Data



Simple reduction



Instance Selection

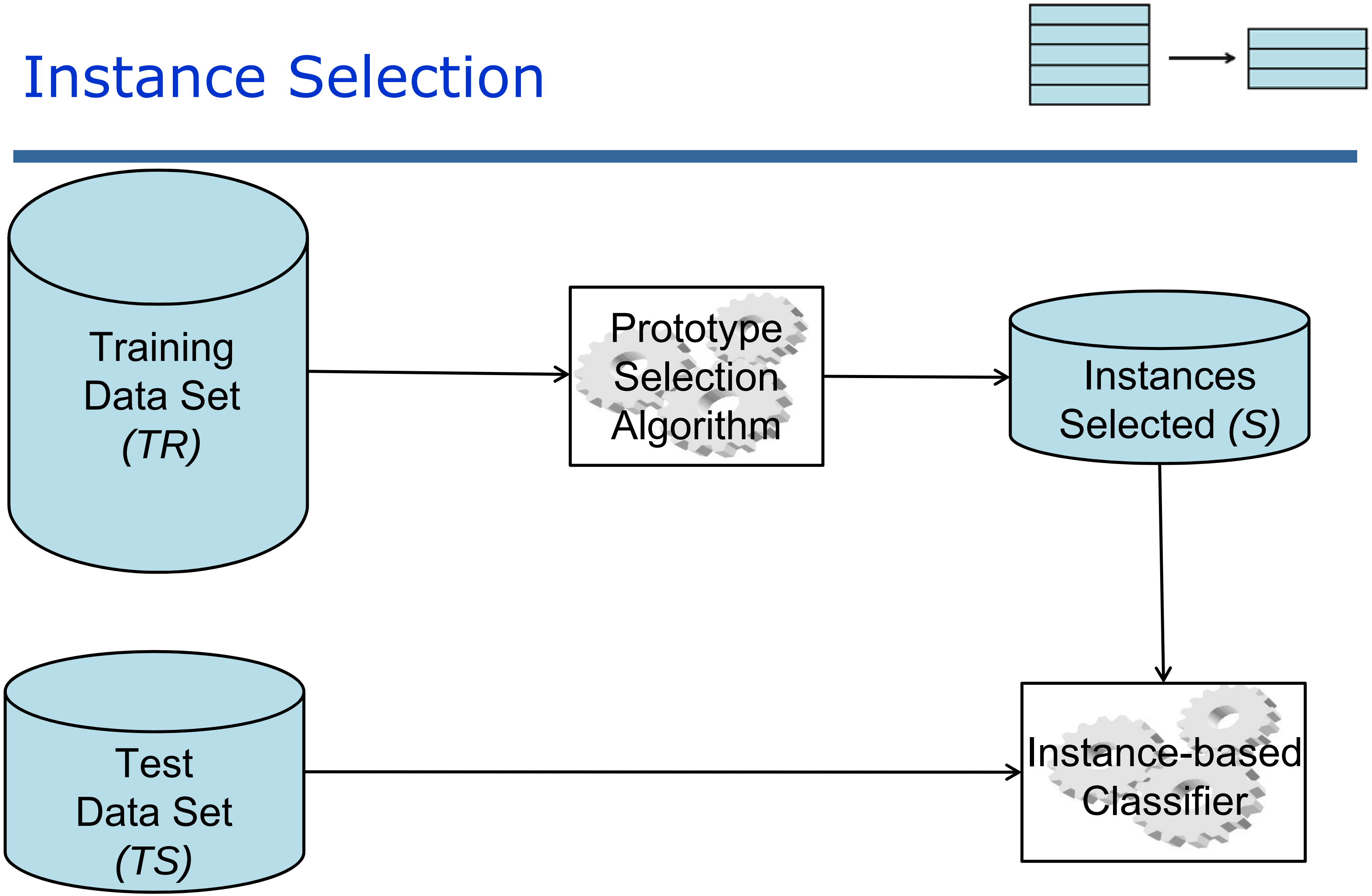
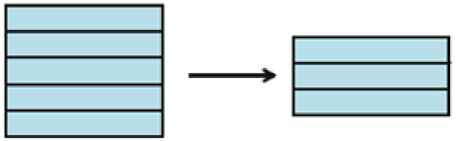


Fig. 8.1 PS process

Instance Selection

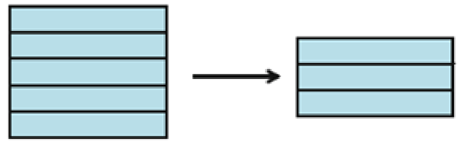


Prototype Selection (instance-based learning)

Properties:

- **Direction of the search:** Incremental, decremental, batch, hybrid or fixed.
- **Selection type:** Condensation, Edition, Hybrid.
- **Evaluation type:** Filter or wrapper.

Instance Selection



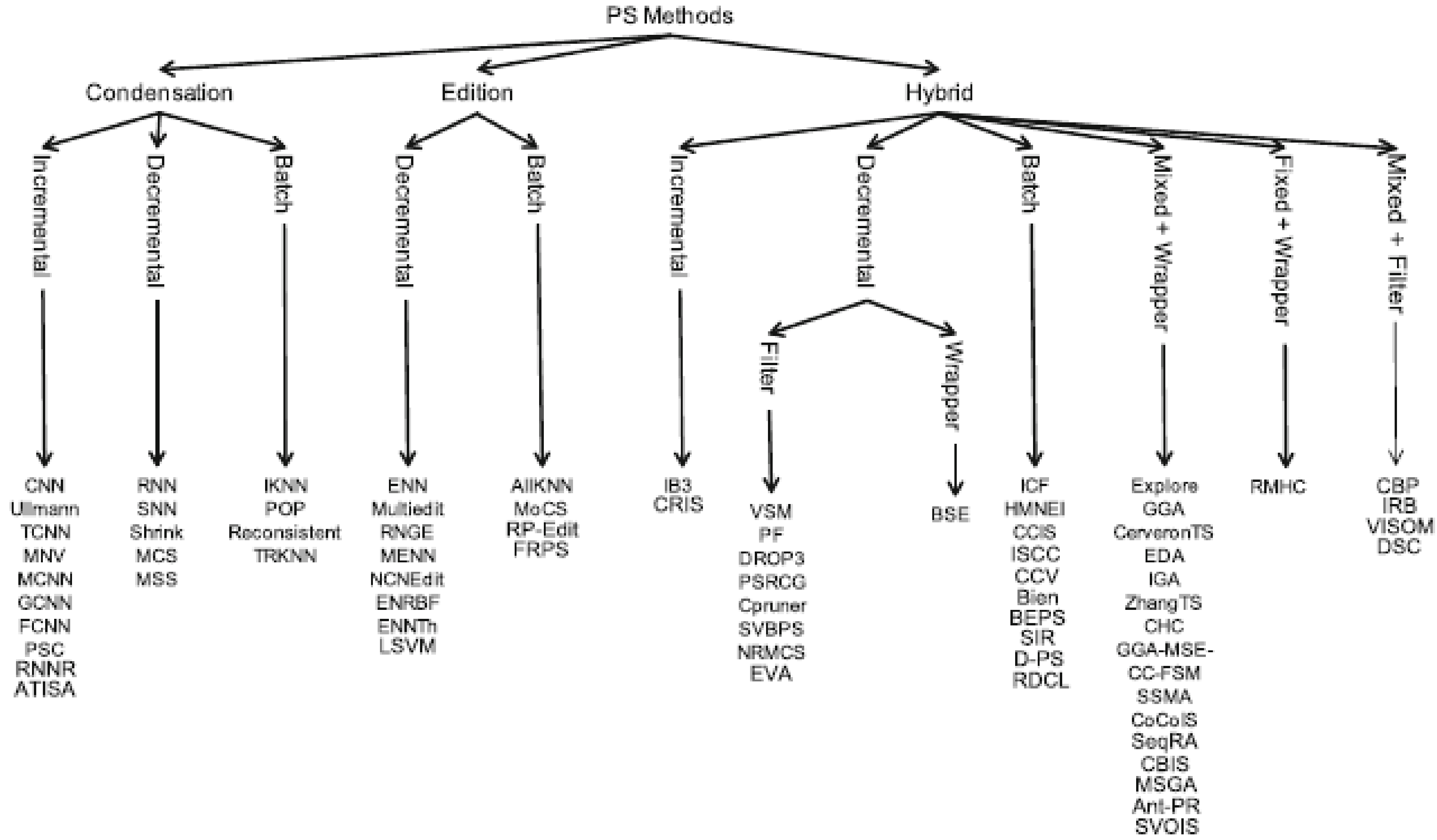
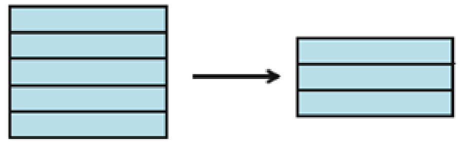


Fig. 8.3 PS taxonomy

Instance Selection



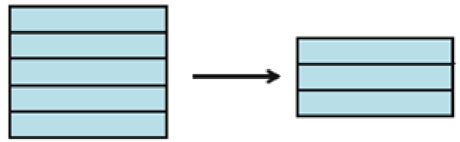
A pair of classical algorithms:

- Classical algorithm of condensation: Condensed Nearest Neighbor (CNN)
  - Incremental
  - It only inserts the misclassified instances in the new subsets.
  - Dependant on the order of presentation.
  - It only retains borderline examples.

Algorithm 10 CNN algorithm.

function CNN(T - training data)
    initialize: $S = \emptyset$
    repeat
        for all $x \in T$ (in random order) **do**
            Find $x' \in S$ s.t. $||x - x'|| = \min_{x^j \in S} ||x - x^j||$
            if $class(x) \neq class(x')$ **then**
                $S = S \cup \{x\}$
            end if
        end for
    until S does not change
    return S
end function

Instance Selection



A pair of classical algorithms:

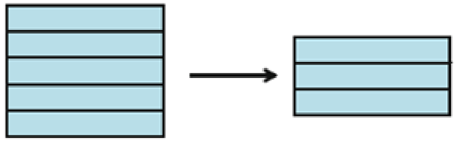
- Classical algorithm for Edition: Edited Nearest Neighbor (ENN)
  - Batch
  - It removes those instances which are wrongly classified by using a k-nearest neighbor scheme (k = 3, 5 or 9).
  - It "smooths" the borders among classes, but also retains the rest of points.

Algorithm 11 ENN algorithm.

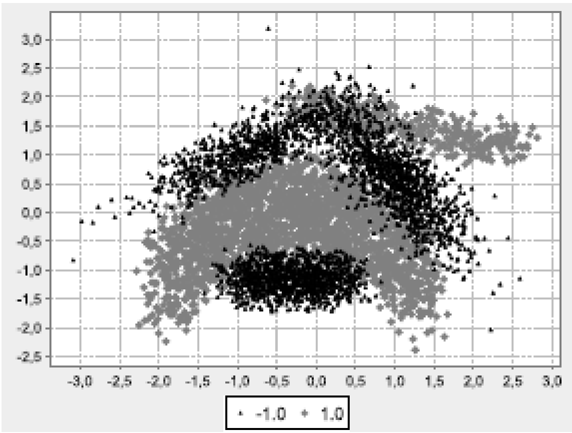
---

function ENN(T - training data, k - number of nearest neighbor)
    initialize: $S = T$
    for all $x \in S$ **do**
        $X' = \emptyset$
        for $i = 1$ to k **do**
            Find $x'_i \in T$ s.t. $x \neq x'_i$ and $||x - x'_i|| = \min_{x^j \in (T \setminus X')} ||x - x^j||$
            $X' = X' \cup \{x'_i\}$
        end for
        if $class(x) \neq majorityClass(X')$ **then**
            $S = S \setminus \{x\}$
        end if
    end for
    return S
end function

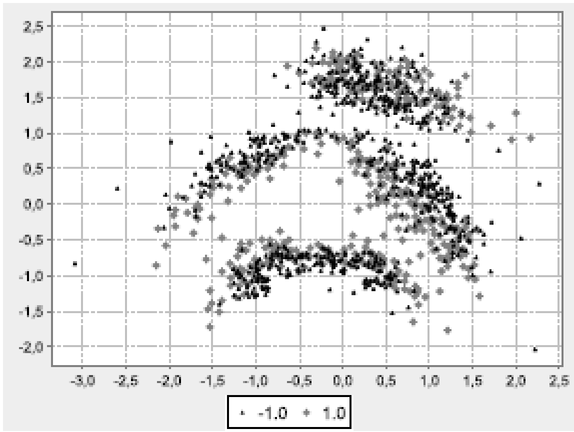
Instance Selection



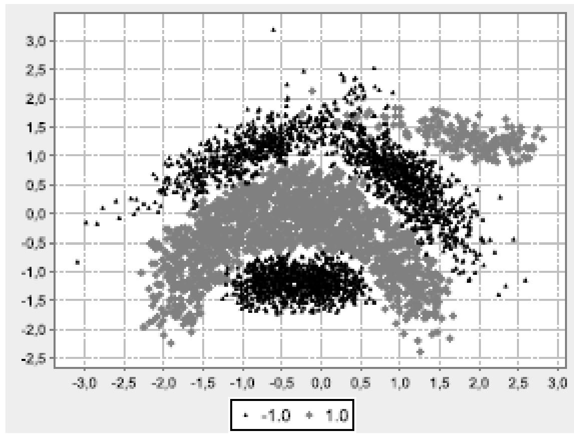
Graphical illustrations:



(a) Banana Original $(0.8751, 0.7476)$



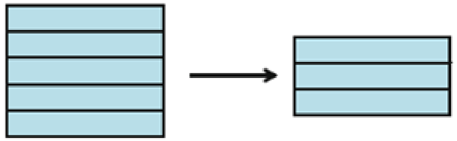
(b) CNN $(0.7729, 0.8664, 0.7304)$



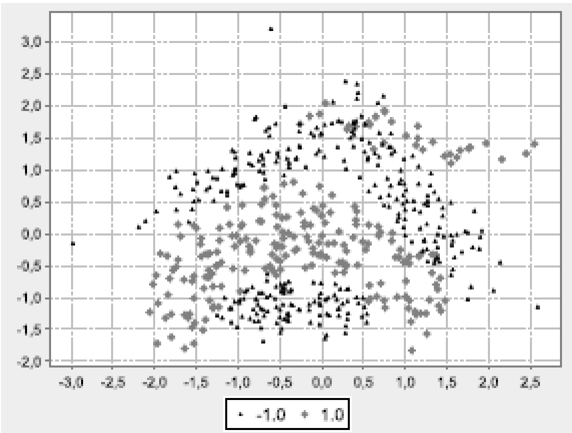
(h) AllKNN $(0.1758, 0.8934, 0.7831)$

Banana data set with 5,300 instances and two classes. Obtained subset with CNN and AllKNN (iterative application of ENN with k=3, 5 y 7).

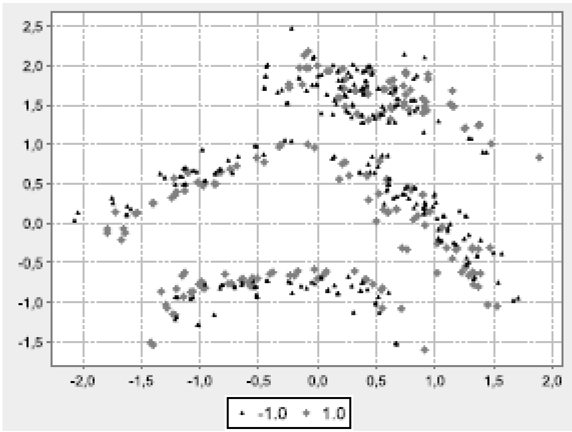
Instance Selection



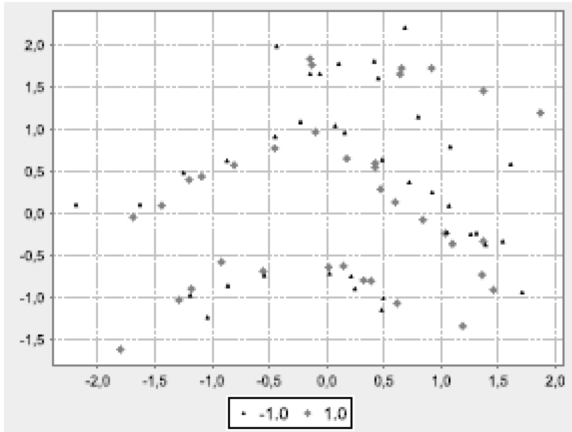
Graphical illustrations:



(k) RMHC $(0.9000, 0.8972, 0.7915)$



(e) DROP3 $(0.9151, 0.8696, 0.7356)$



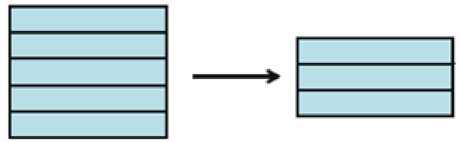
(l) SSMA $(0.9879, 0.8964, 0.7900)$

RMHC is an adaptive sampling technique based on local search with a fixed final rate of retention.

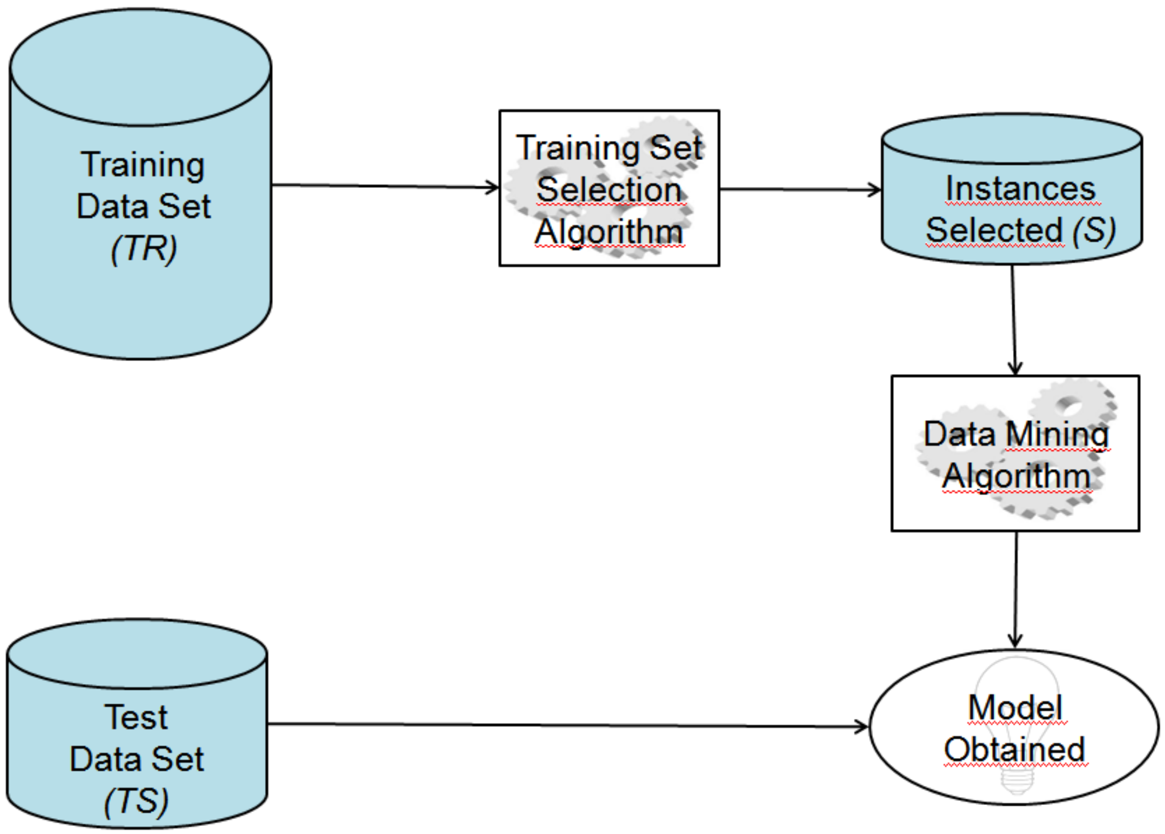
DROP3 is the most-known hybrid technique very use for kNN.

SSMA is an evolutionary approach based on memetic algorithms..

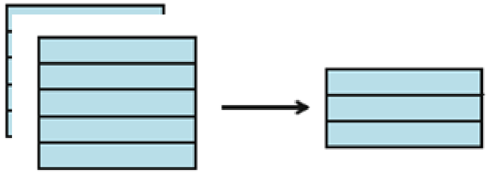
Instance Selection



Training Set Selection



Example Instance Selection and Decision Tree modeling

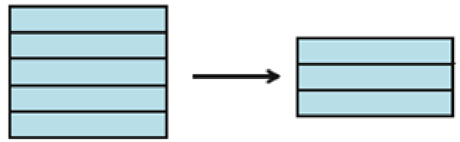


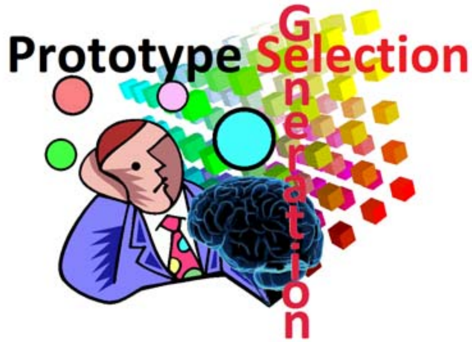
Kdd Cup'99. Strata Number: 100

	No. Rules	% Reduction	C4.5	
			%Ac Trn	%Ac Test
C4.5	252		99.97%	99.94%
Cnn Strat	83	81.61%	98.48%	96.43%
Drop1 Strat	3	99.97%	38.63%	34.97%
Drop2 Strat	82	76.66%	81.40%	76.58%
Drop3 Strat	49	56.74%	77.02%	75.38%
Ib2 Strat	48	82.01%	95.81%	95.05%
Ib3 Strat	74	78.92%	99.13%	96.77%
Icf Strat	68	23.62%	99.98%	99.53%
CHC Strat	**9**	**99.68%**	98.97%	**97.53%**

Bibliography: J.R. Cano, F. Herrera, M. Lozano, **Evolutionary Stratified Training Set Selection for Extracting Classification Rules with Trade-off Precision-Interpretability**. *Data and Knowledge Engineering 60 (2007) 90-108, doi:10.1016/j.datak.2006.01.008*.

Instance Selection





WEBSITE:
http://sci2s.ugr.es/pr/index.php
Bibliography:

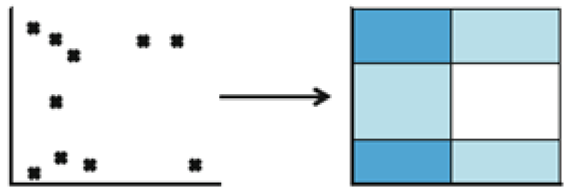
S. García, J. Derrac, J.R. Cano and F. Herrera,
Prototype Selection for Nearest Neighbor Classification: Taxonomy and Empirical Study.
IEEE Transactions on Pattern Analysis and Machine Intelligence 34:3 (2012) 417-435 doi: 10.1109/TPAMI.2011.142

S. García, J. Luengo, F. Herrera. **Data Preprocessing in Data Mining**, Springer, 15, 2015

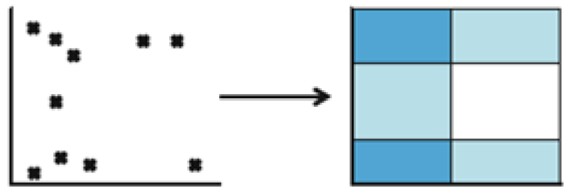
Source Codes (Java):

KEEL

Discretization



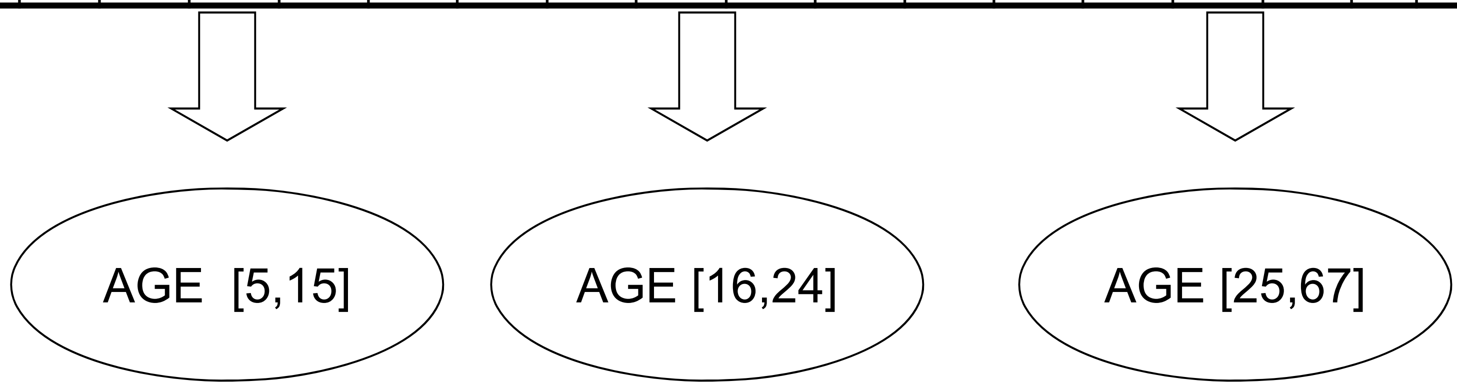
- Discrete values are very useful in Data Mining.
- They represent more concise information, theay are easier to understand and closer to the representation of knowledge.
- The discretization is focused on the transformation of continuous values with an order among in nominal/categorical values without ordering. It is also a quantification of numerical attributes.
- Nominal value are within a finite domain, so they are also considered as a data reduction technique.



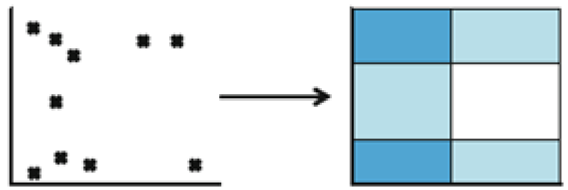
Discretization

- Divide the range of numerical (continuos or not) attributes into intervals.
- Store the labels of the intervales.
- Is crucial for association rules and some classification algorithms, which only accepts discrete data.

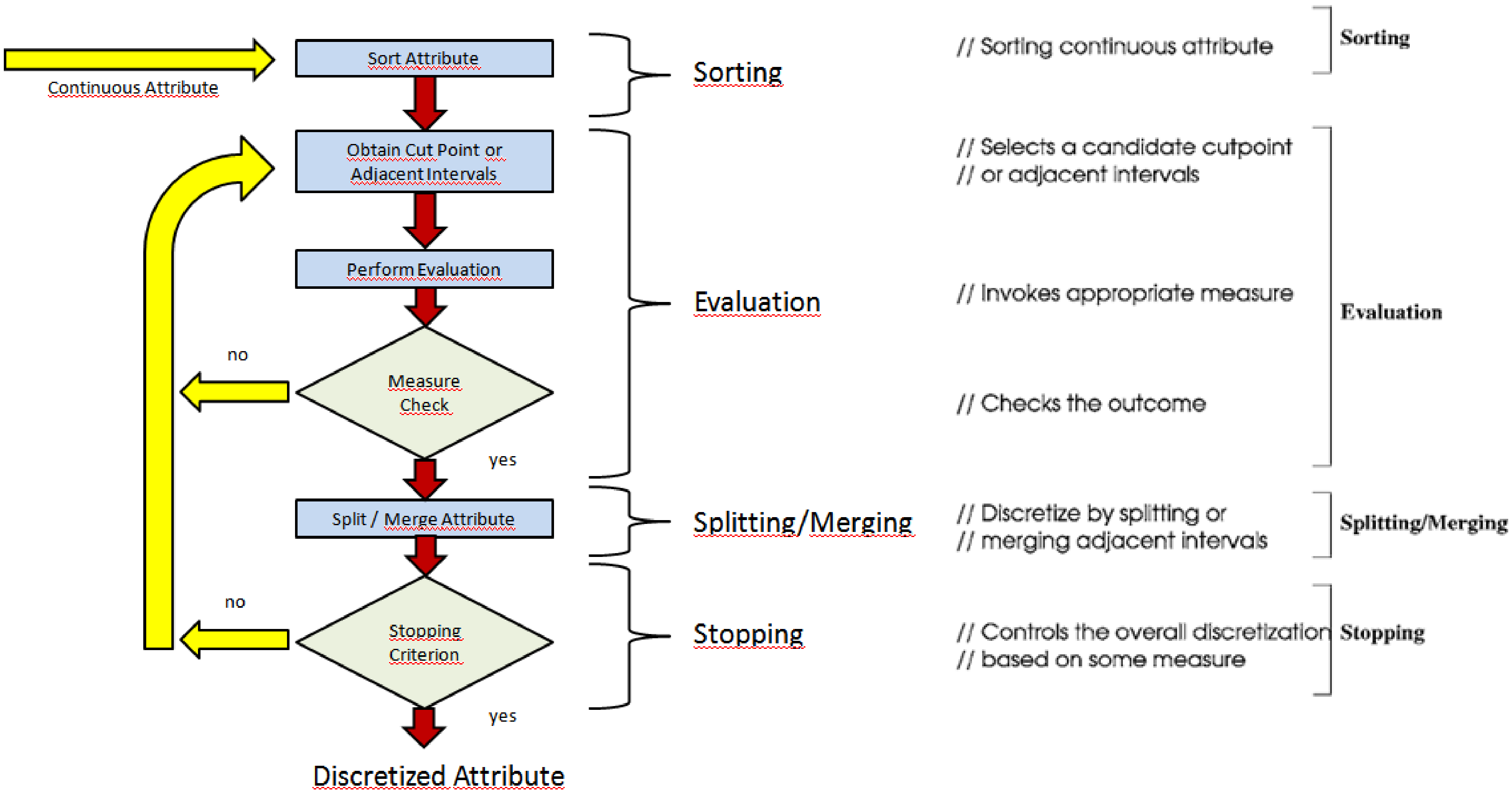
Age	5	6	6	9	...	15	16	16	17	20	...	24	25	41	50	65	...	67
Owner of a Car	0	0	0	0	...	0	1	0	1	1	...	0	1	1	1	1	...	1



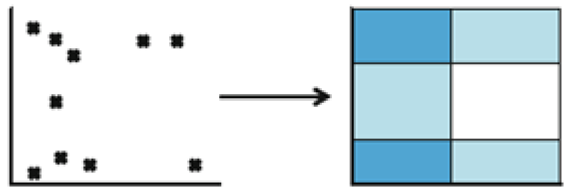
Discretization



Stages in the discretization process



Discretization



- **Discretization has been developed in several lines according to the neccesities:**

- Supervised vs. unsupervised: Whether or not they consider the objective (class) attributes.

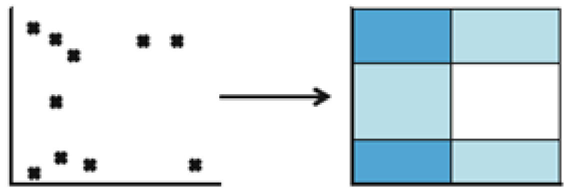
- Dinamical vs. Static: Simultaneously when the model is built or not.

- Local vs. Global: Whether they consider a subset of the instances or all of them.

- Top-down vs. Bottom-up: Whether they start with an empty list of cut points (adding new ones) or with all the possible cut points (merging them).

- Direct vs. Incremental: They make decisions all together or one by one.

Discretization



- **Unsupervised algorithms:**
  - Equal width
  - Equal frequency
  - Clustering …..

- Supervised algorithms:

  - Entropy based [Fayyad & Irani 93 and others]

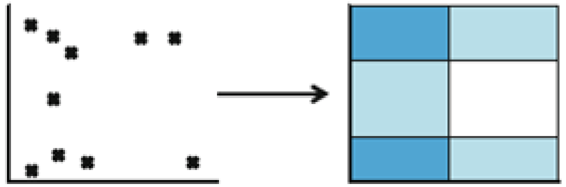
    [Fayyad & Irani 93] U.M. Fayyad and K.B. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. *Proc. 13th Int. Joint Conf. AI (IJCAI-93)*, 1022-1027. Chamberry, France, Aug./ Sep. 1993.

  - Chi-square [Kerber 92]

    [Kerber 92] R. Kerber. ChiMerge: Discretization of numeric attributes. *Proc. 10th Nat. Conf. AAAI,* 123-128. 1992.

  - … (lots of proposals)

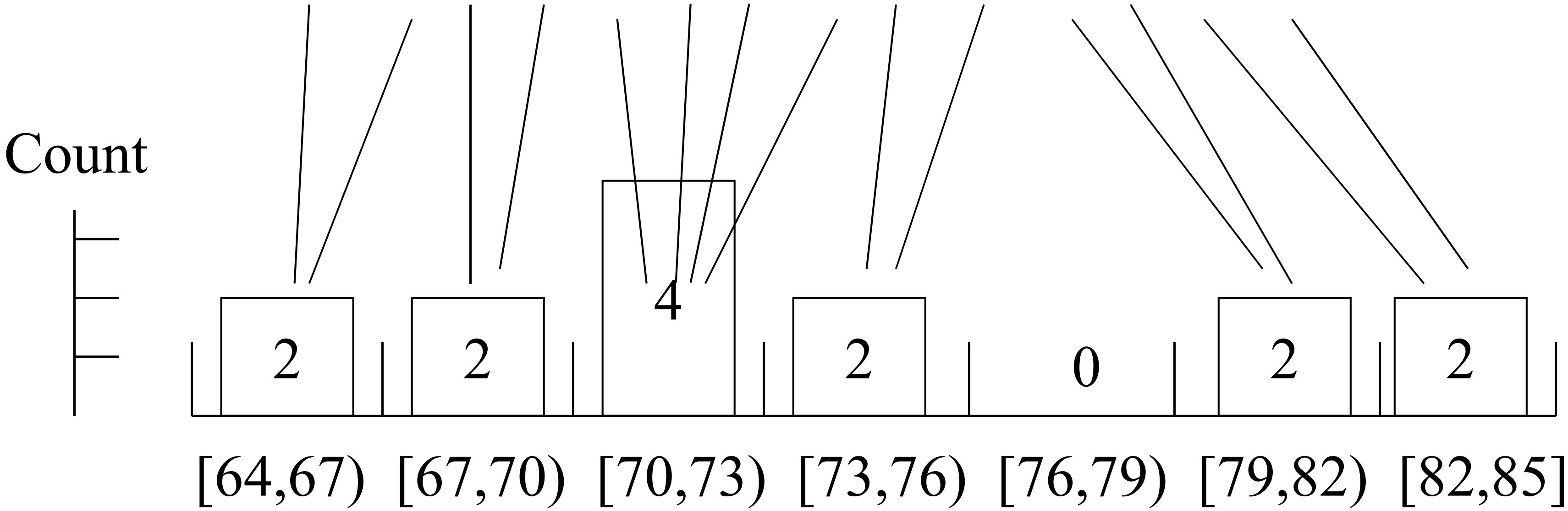
Bibliography: S. García, J. Luengo, José A. Sáez, V. López, F. Herrera, A Survey of Discretization Techniques: Taxonomy and Empirical Analysis in Supervised Learning.
IEEE Transactions on Knowledge and Data Engineering 25:4 (2013) 734-750, *doi: 10.1109/TKDE.2012.35*.



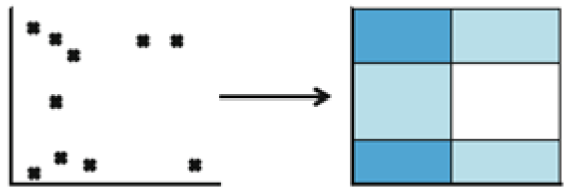
Discretization

Example Discretization: Equal width

Temperature:
64 65 68 69 70 71 72 72 75 75 80 81 83 85



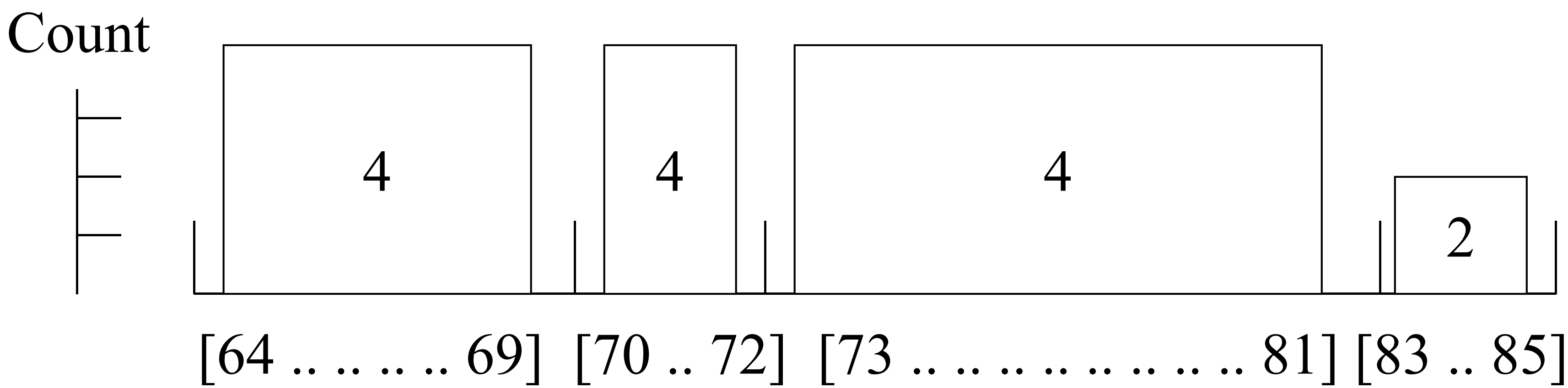
Equal width



Discretization

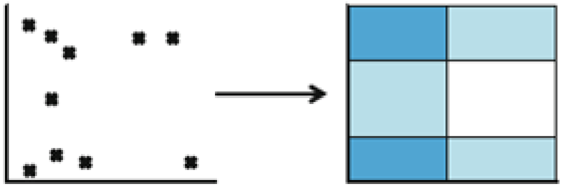
Example discretization: Equal frequency

Temperature
64 65 68 69 70 71 72 72 75 75 80 81 83 85



Equal frequency (height) = 4, except for the last box

Discretization



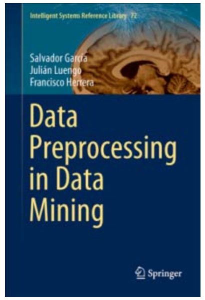
- *Which discretizer will be the best?.*

- As usual, it will depend on the application, user requiriments, etc.
- Evaluation ways:
  - Total number of intervals
  - Number of inconsistencies
  - Predictive accuracy rate of classifiers

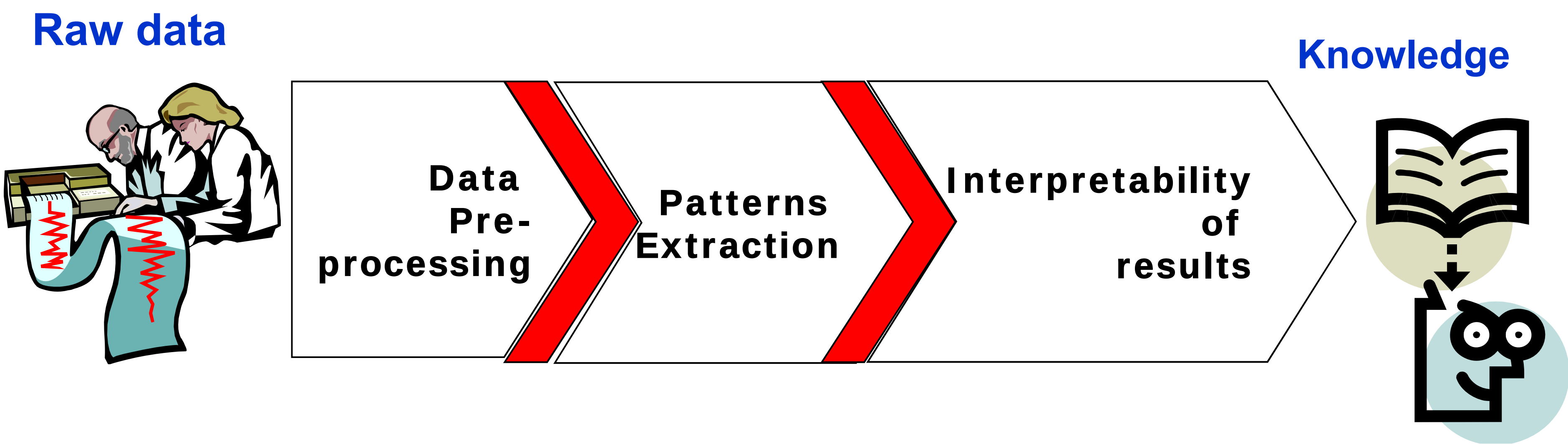
Data Preprocessing in Data Mining

1. Introduction. Data Preprocessing
2. Integration, Cleaning and Transformations
3. Imperfect Data
4. Data Reduction
5. Final Remarks

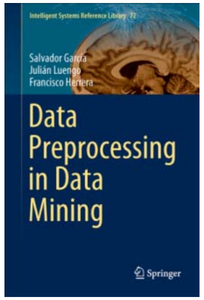
Final Remarks



Data preprocessing is a necessity when we work with real applications.

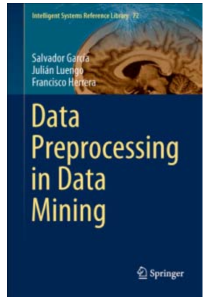


Final Remarks



Advantage: Data preprocessing allows us to apply Learning/Data Mining algorithms easier and quicker, obaining more quality models/patterns in terms of accuracy and/or interpretability.

Final Remarks



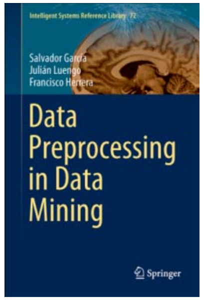
Advantage: Data preprocessing allows us to apply Learning/Data Mining algorithms easier and quicker, obaining more quality models/patterns in terms of accuracy and/or interpretability.

A drawback: Data preprocessing is not a structured area with a specific methodology for understand the suitability of preprocessing algorithms for managing a new problems.

Every problem can need a different preprocessing process, using different tools.

The design of automatic processes of use of the different stages/techniques is one of the data mining challenges.

Final Remarks

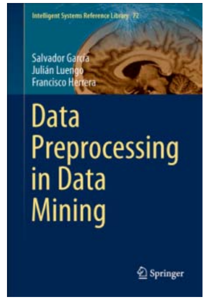


KEEL software for Data Mining (knowledge extraction based on evolutionary learning) includes a data preprocessing module (feature selection, missing data imputation, instance selection, discretization, …)

http://www.keel.es/

...rithms included in KEEL (484)		
...ily	Subfamily	
Data Preprocessing (98)	Discretization (30)	
	Feature Selection (25)	Feature Selection (22)
		Evolutionary Feature Selection (3)
	Training Set Selection (16)	Training Set Selection (12)
		Evolutionary Training Set Selection (4)
	Missing Values (15)	
	Transformation (4)	
	Data Complexity (1)	
	Noisy Data Filtering (7)	

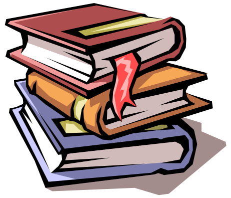
Final Remarks

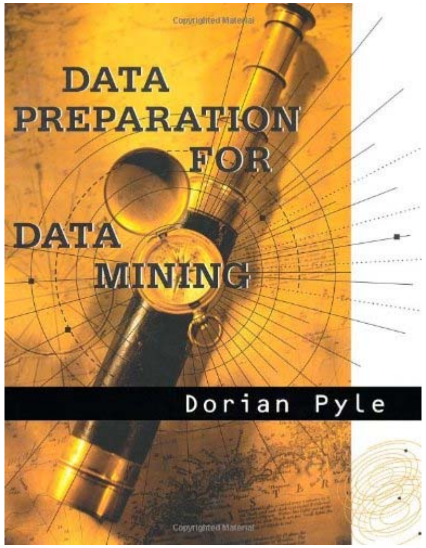


Summary

- Data preprocessing is a big issue for data mining
- Data processing includes
  - Data preparation: cleaning, imperfect data, transformation …
  - Data reduction and data transformation
- A lot a methods have been developed but still an active area of research
- The cooperation between data mining algorithms and data preparation methods is an interesting/active area.

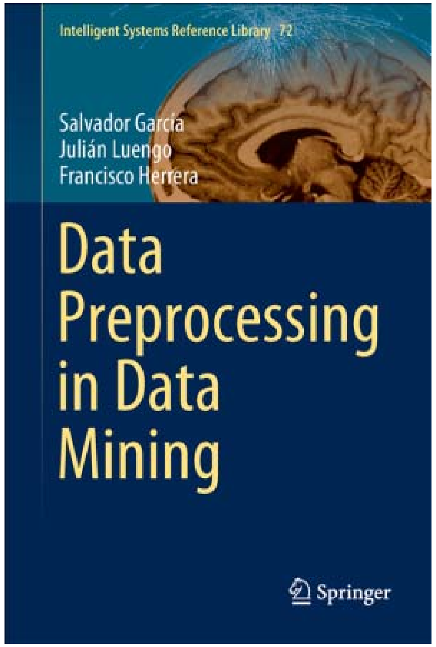
Bibliography





Dorian Pyle
Morgan Kaufmann, Mar 15, 1999

"Good data preparation is key to produce valid and reliable models"



S. García, J. Luengo, F. Herrera
Data Preprocessing in Data Mining
Springer, 15, 2015

Data Preprocessing