TÉCNICO LISBOA

Intelligent Systems

Susana M. Vieira

Universidade de Lisboa, Instituto Superior Técnico

IS4, Center of Intelligent Systems, IDMEC, LAETA, Portugal
{susana.vieira}@tecnico.ulisboa.pt

TÉCNICO LISBOA

Feature Selection and Knowledge Discovery

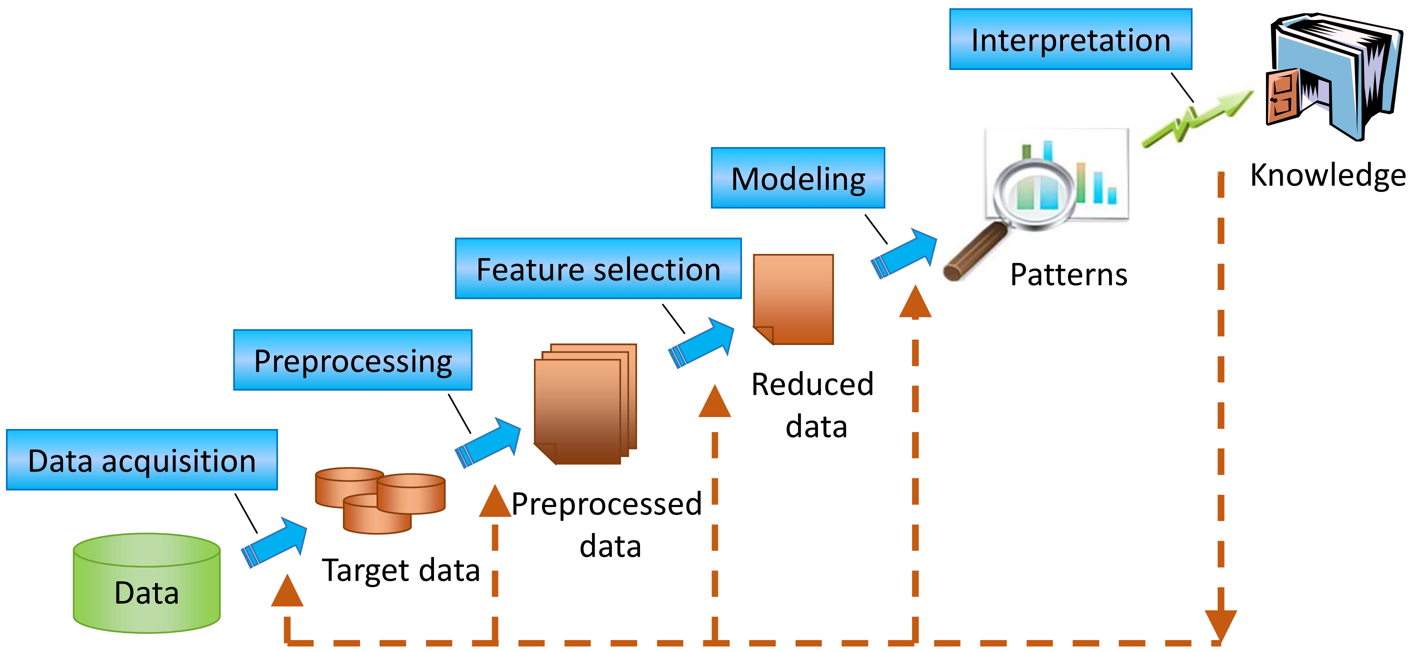
SI7 – Intelligent data analysis, KDD, Feature selection, Feature extraction

Guyon, I., Gunn, S., Nikravesh, M., Zadeh, L.A.. ***Feature Extraction: Foundations and Applications***. 2006.

J. Li, K. Cheng, S. Wang, F. Morstatter, T. Robert, J. Tang, and H. Liu. ***Feature selection: A data perspective***. 2016

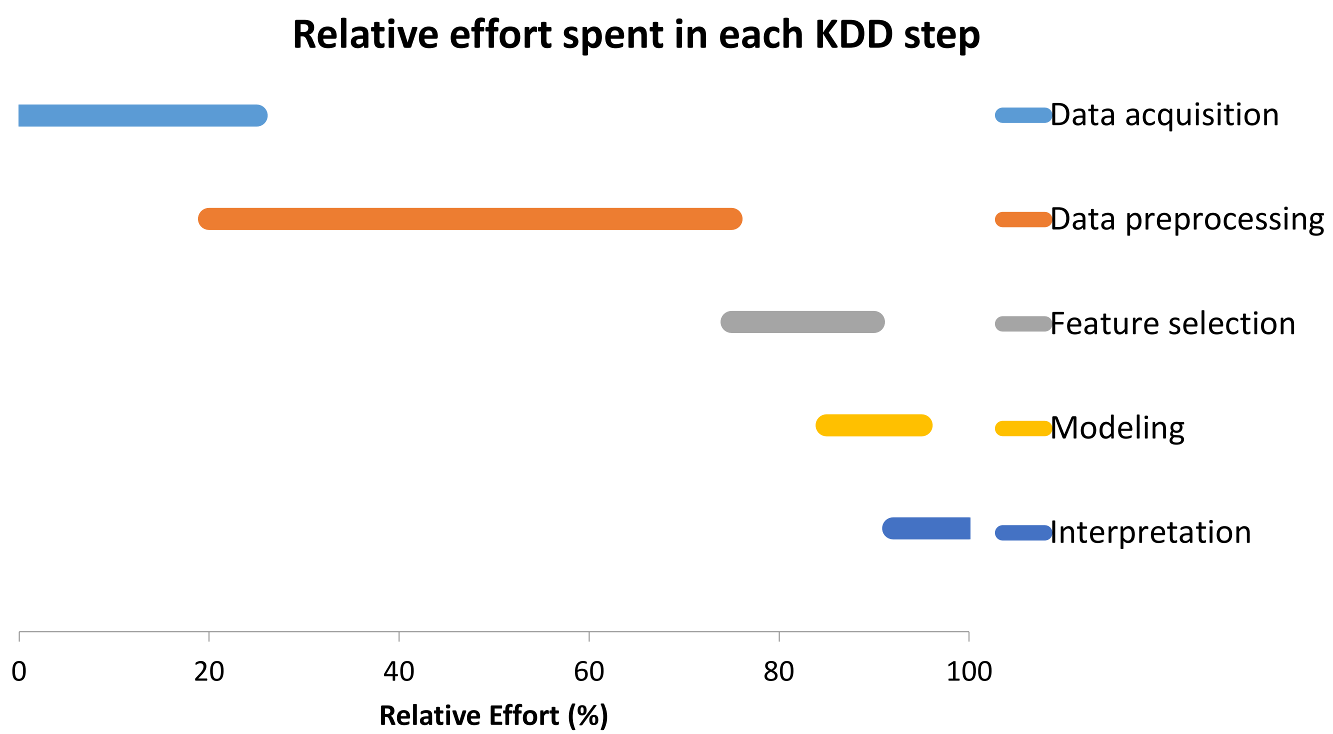
Michael R. Berthold and Christian Borgelt. ***Guide to Intelligent Data Analysis: How to Intelligently Make Sense of Real Data***. 2010.

Knowledge Data Discovery

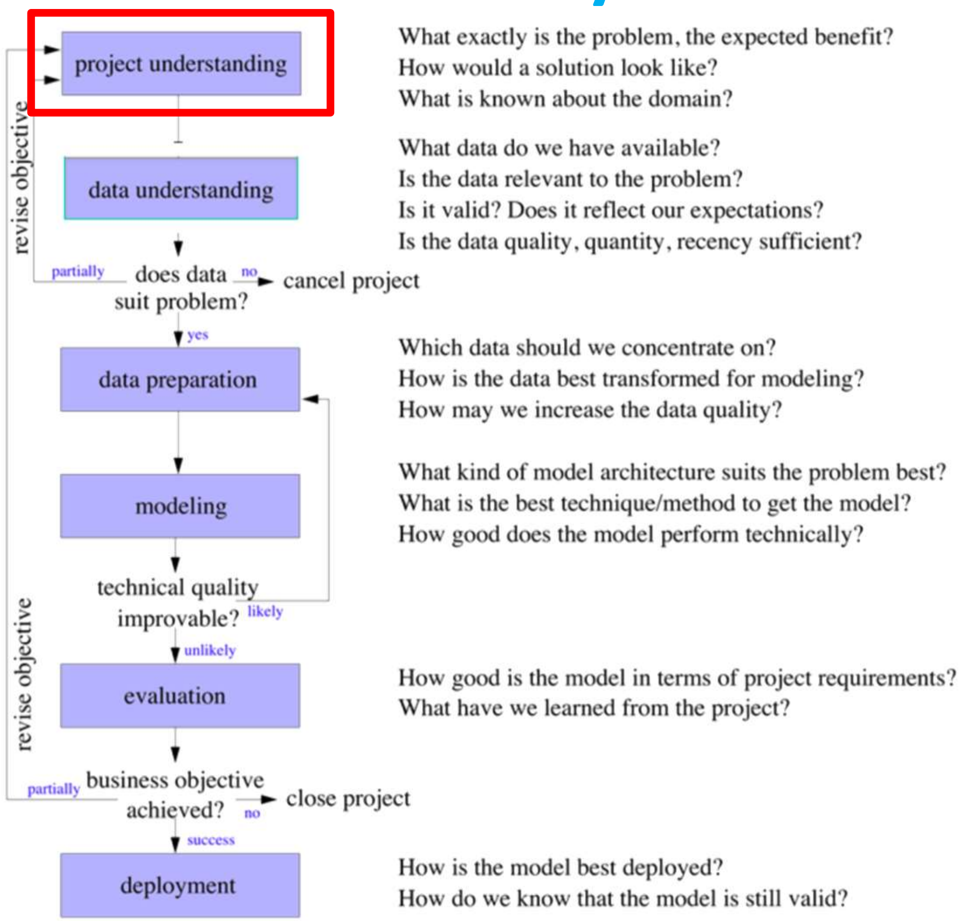


Based on "G. Piatetsky-Shapiro U. Fayyad and P. Smyth. From data mining to knowledge discovery in databases. *Artificial Intelligence Magazine*, 17(3):37-54, 1996."

Knowledge Discovery in Databases



Intelligent Data Analysis



Determine the project objective

Problems faced in data analysis

problem source	project owner perspective	analyst perspective
communication	project owner does not understand the technical terms of the analyst	analyst does not understand the terms of the domain of the project owner
lack of understanding	project owner was not sure what the analyst could do or achieve models of analyst were different from what the project owner envisioned	analyst found it hard to understand how to help the project owner
organization	requirements had to be adopted in later stages as problems with the data became evident	project owner was an unpredictable group (not so concerned with the project)

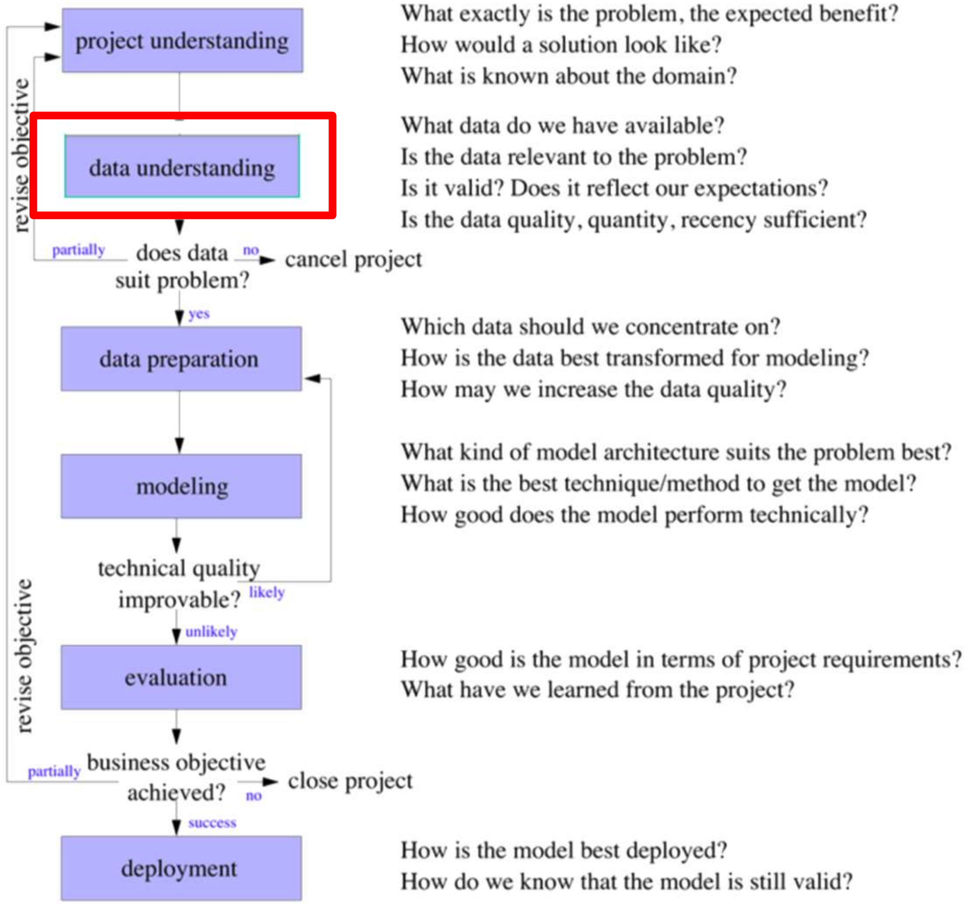
TÉCNICO LISBOA

Determine the project objective

- **Determine data mining tasks**
  - (classification, regression, cluster analysis, finding associations, deviation analysis,…)
- **Specify the requirements for the models**
- **Determine analysis goals**
  - Interpretability
  - Reproducibility/stability
  - Model
  - Flexibility/adequacy
  - Runtime
  - Interestingness and use of expert knowledge

TÉCNICO LISBOA

Intelligent Data Analysis

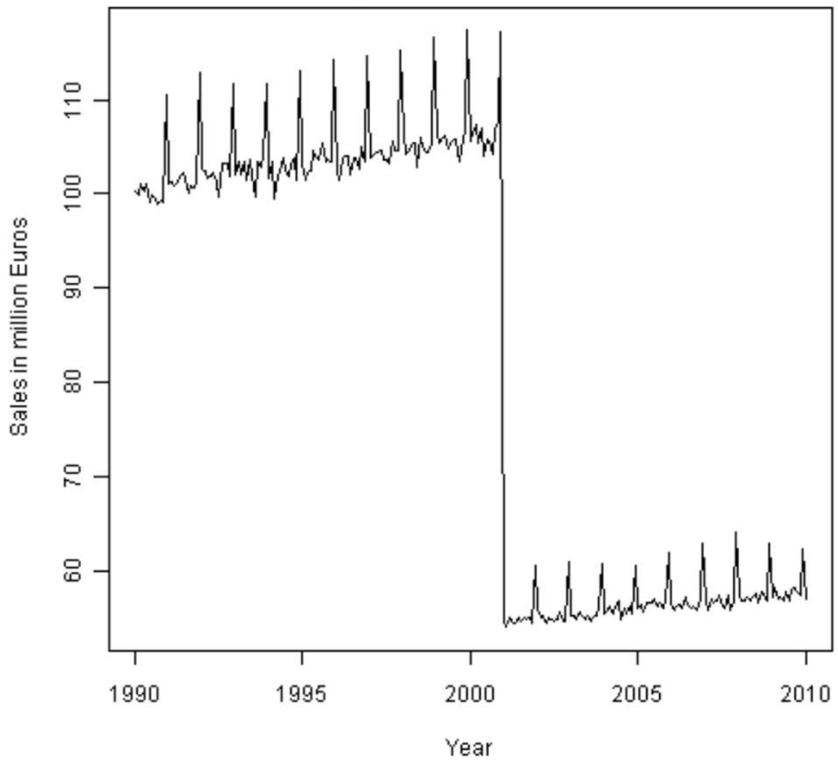


Questions in data understanding

- **Goal: gain insight in your data** with respect to your project goals
- **Find answers to the questions:**
  - What kind of attributes do we have?
  - How is the data quality?
  - Does a visualization helps?
  - Are attributes correlated?
  - What about outliers?
  - How are missing values handled?

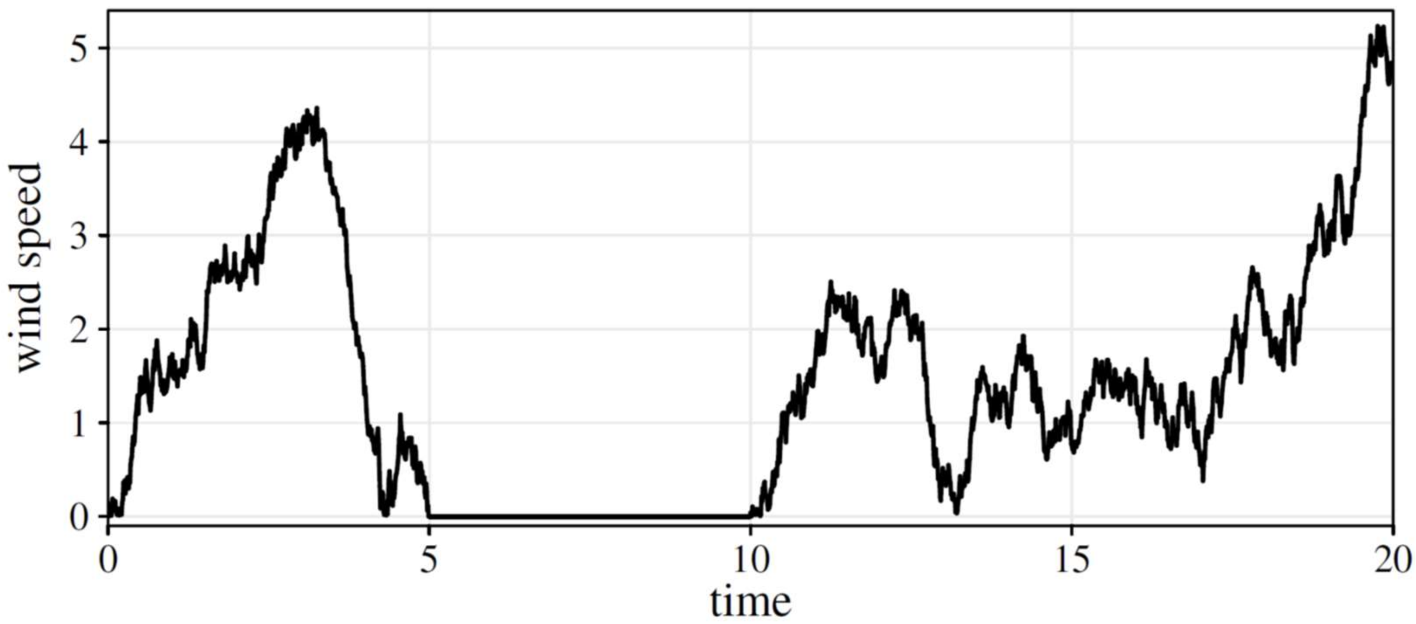
Data visualization

- **There is no excuse for failing to plot and look**



Data visualization

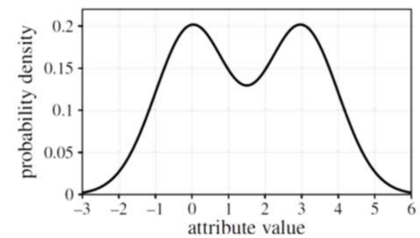
- **There is no excuse for failing to plot and look**

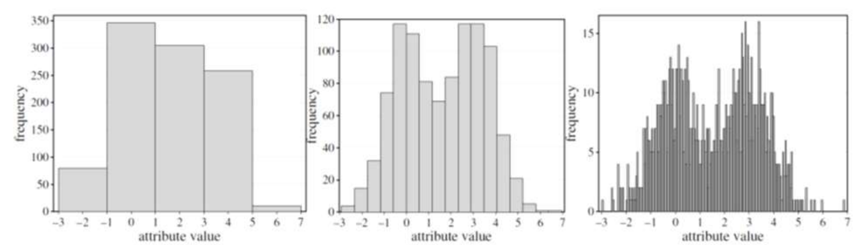


Hidden missing values

Data understanding checklist: Must do!

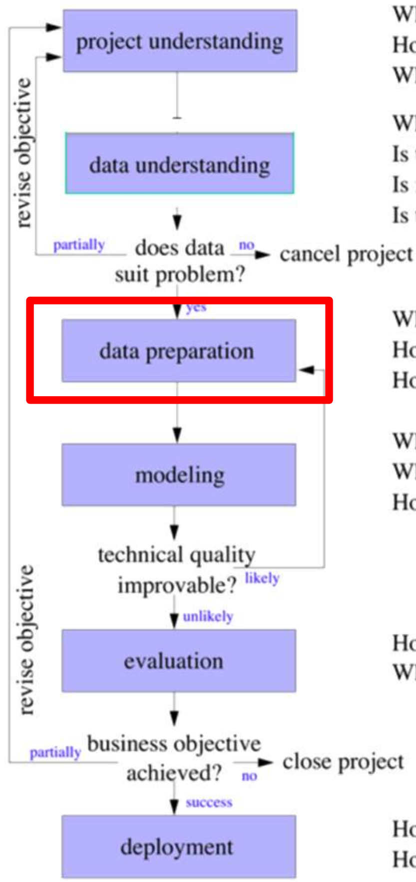
- **Check the distributions for each attribute**
  - (unexpected properties like outliers, correct domains, correct medians)





- **Check correlations or dependencies between pairs of attributes**

Intelligent Data Analysis



What exactly is the problem, the expected benefit?
How would a solution look like?
What is known about the domain?

What data do we have available?
Is the data relevant to the problem?
Is it valid? Does it reflect our expectations?
Is the data quality, quantity, recency sufficient?

Which data should we concentrate on?
How is the data best transformed for modeling?
How may we increase the data quality?

What kind of model architecture suits the problem best?
What is the best technique/method to get the model?
How good does the model perform technically?

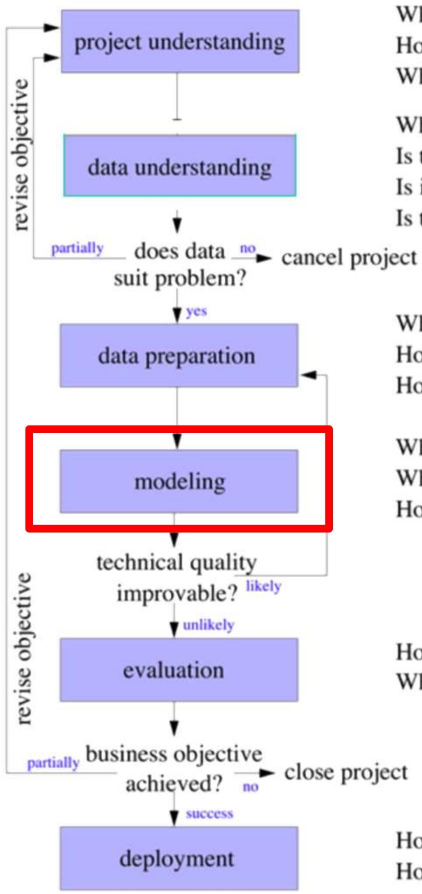
How good is the model in terms of project requirements?
What have we learned from the project?

How is the model best deployed?
How do we know that the model is still valid?

Data understanding vs Data preparation

- **Data understanding** provides general information about the data
  - Existence and character of missing values
  - Outliers
  - Character of attributes and dependencies between attributes.
- **Data preparation** uses this information to select attributes
  - Reduce the dimension of the data set
  - Select records
  - Treat missing values and outliers
  - Integrate, unify and transform data; improve data quality.

Intelligent Data Analysis



What exactly is the problem, the expected benefit?
How would a solution look like?
What is known about the domain?

What data do we have available?
Is the data relevant to the problem?
Is it valid? Does it reflect our expectations?
Is the data quality, quantity, recency sufficient?

Which data should we concentrate on?
How is the data best transformed for modeling?
How may we increase the data quality?

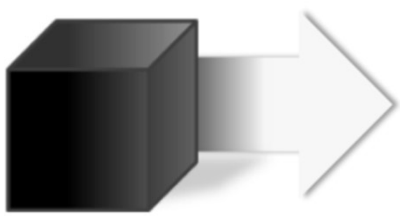
What kind of model architecture suits the problem best?
What is the best technique/method to get the model?
How good does the model perform technically?

How good is the model in terms of project requirements?
What have we learned from the project?

How is the model best deployed?
How do we know that the model is still valid?

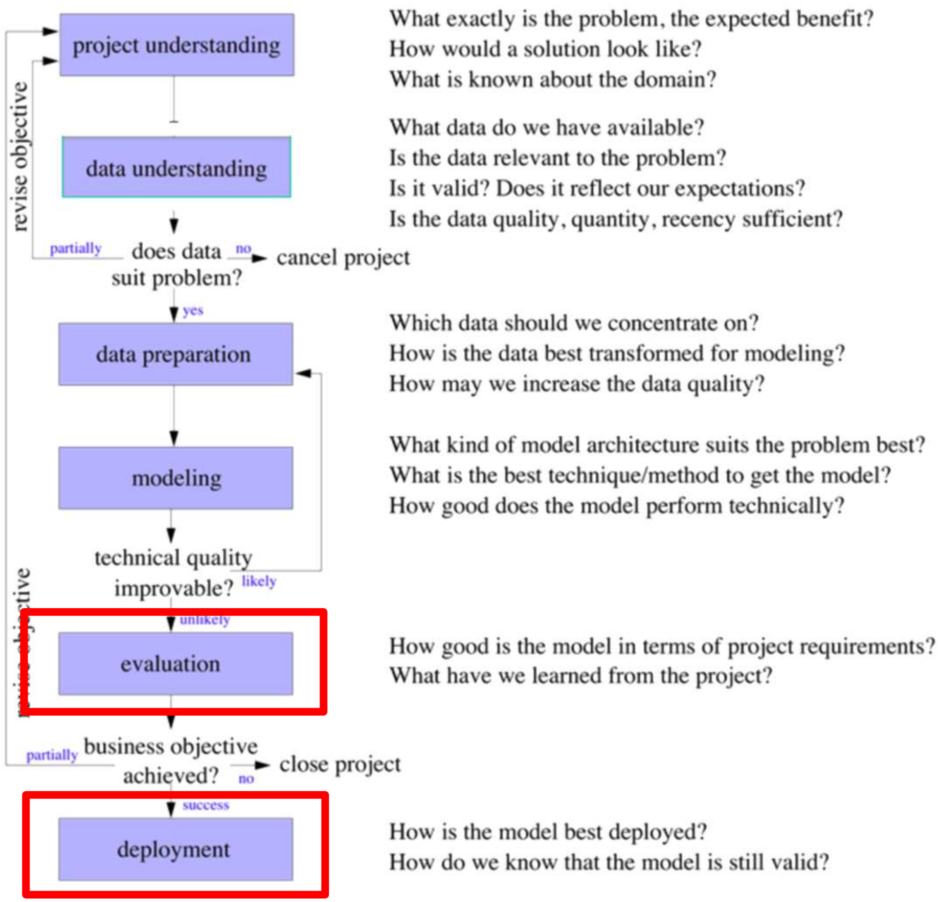
Model: requirements

- **Simplicity**
  - **Occam's razor**: Choose the simplest model that still "explains" the data.
  - Or: Numquam ponenda est pluralitas sine necessitate
    - = [Plurality must never be posited without necessity]
  - Easier to understand
  - Lower complexity
  - Avoid overfitting

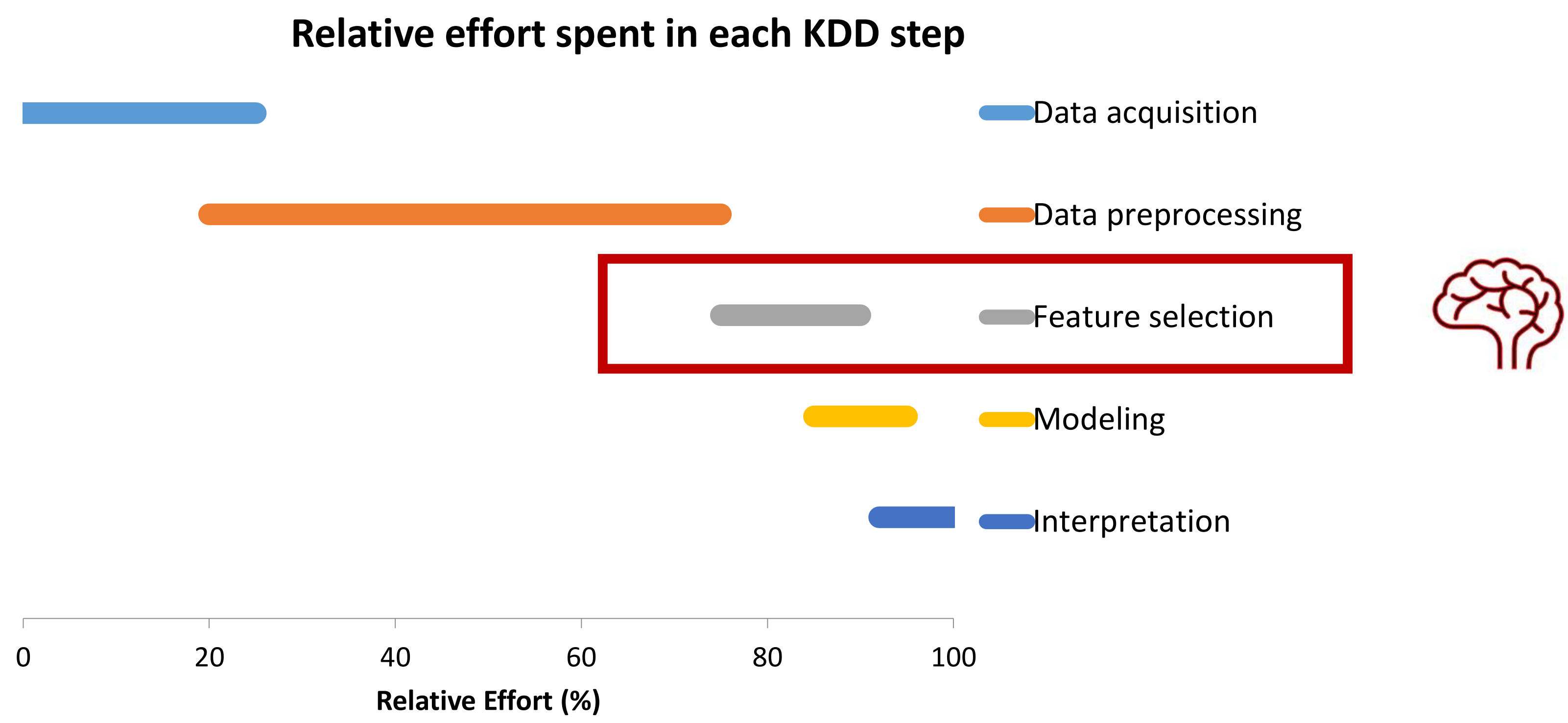


- **Interpretability**
  - Black-Boxes are mostly not a proper choice
  - But: They can result in a very good accuracy (e.g. NN or DNN)

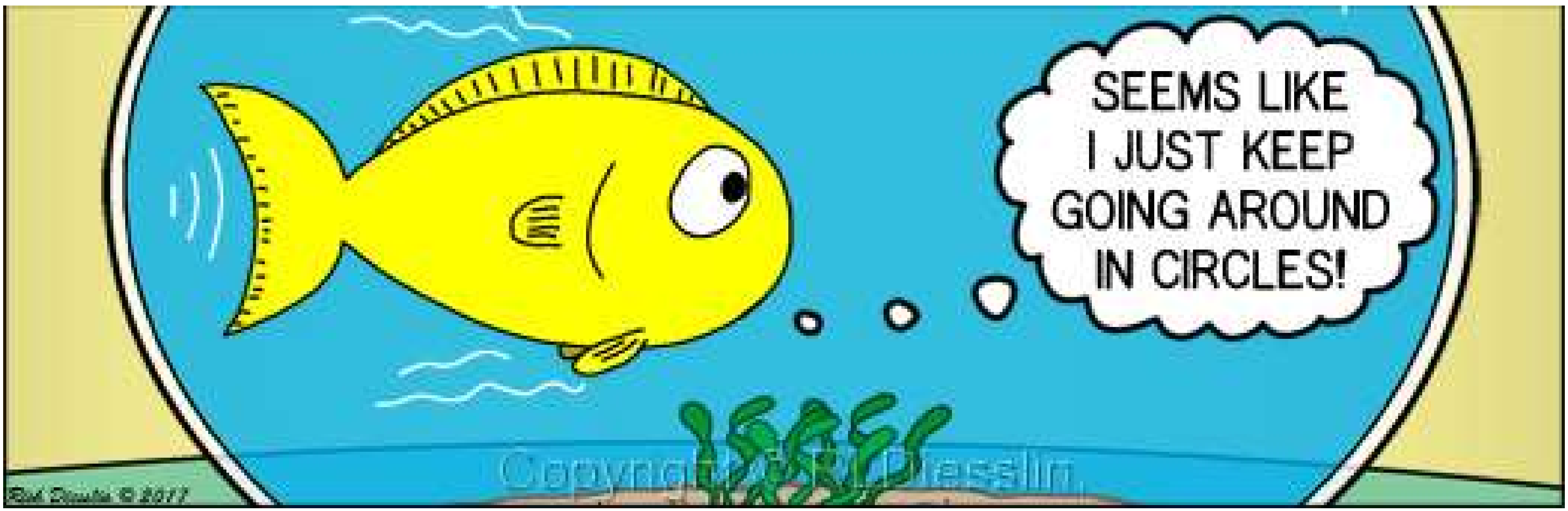
Intelligent Data Analysis



Knowledge Discovery in Databases



Feature Selection



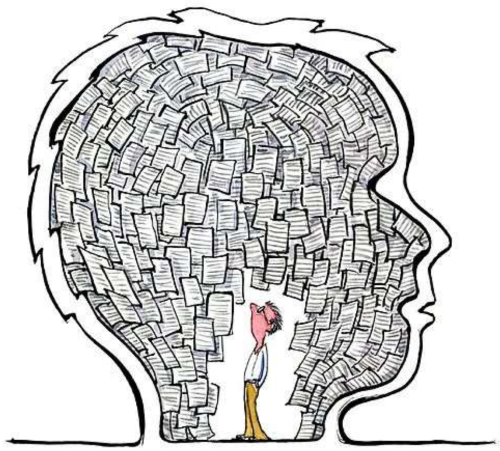
Why feature selection?

- The information about the target class is **inherent in the variables.**
- Naive **theoretical** view - more features
  - More information
  - More discrimination power.
- **In practice** - many reasons why this

is not the case.

Practical problems

- Many explored domains have hundreds to **tens of thousands of variables/features** with many irrelevant and redundant ones.
- In domains with many features the underlying probability distribution can be very complex and very hard to estimate (e.g. dependencies between variables).

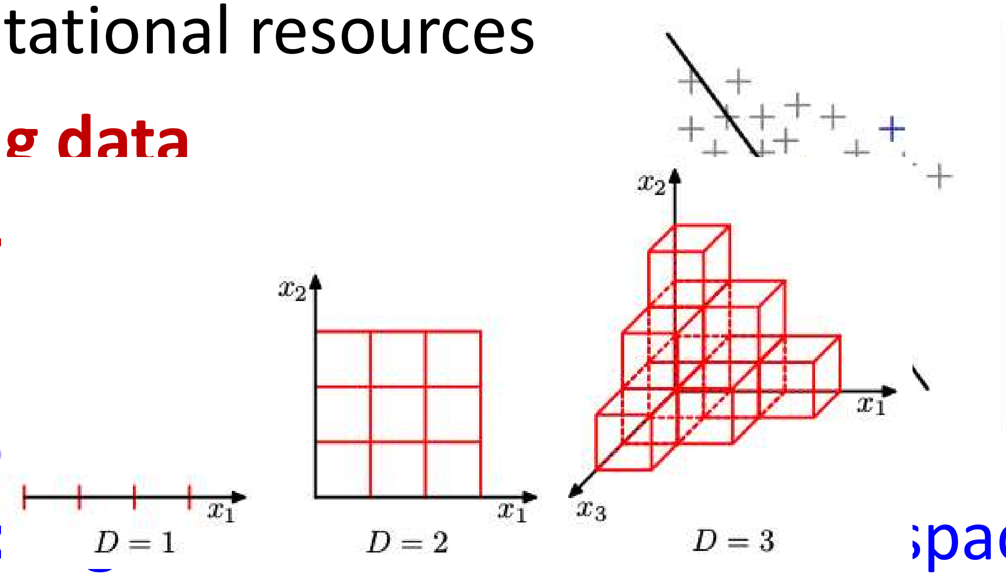


TÉCNICO LISBOA

Practical problems

- Irrelevant and redundant features can “**confuse**” learners
- Limited computational resources
- **Limited training data**
- **Curse of dimer**

➢ In many cases ... carding
variables is ma...
mapping/samp... space

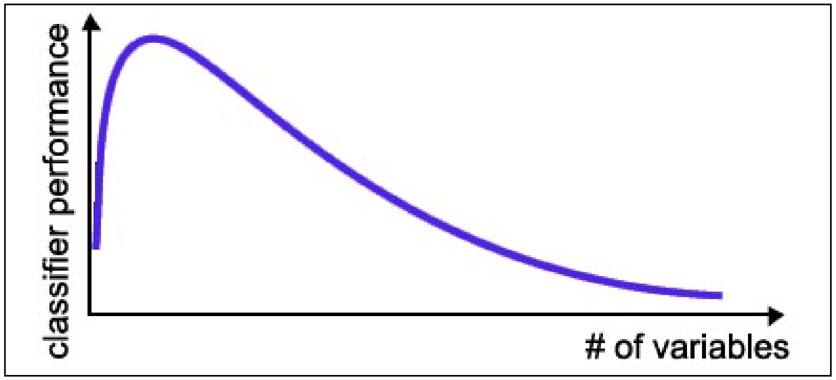


1D: 3 regions, 2D: 3^2 regions, 3D: 3^3, 1000D: hopeless!

TÉCNICO LISBOA

Practical problems

- The required number of samples (to achieve the same accuracy) grows **exponentially** with the number of variables.
- In practice: number of training examples is fixed.
  - Classifier performance usually degrade for a large number of features:



Real-world example

Gene selection from microarray data:

- **Variables:**
  gene expression coefficients corresponding to the amount of mRNA in a patient's sample (e.g. tissue biopsy)
- **Task:** Separate healthy patients from cancer patients
  - Usually there are only about **100 examples** (patients) available for training and testing
  - Number of variables in the raw data: **6.000 – 60.000**
  - Does this work? ([a])

[a] C. Ambroise, G.J. McLachlan: Selection bias in gene extraction on the basis of microarray gene-expression data. *PNAS* Vol. 99 6562-6566(2002)

Feature selection

- What is feature selection?

  Remove features X(*i*) to improve (or least degrade) prediction of Y.

- **Advantages**:
  - Feature selection specify the most relevant features
  - Collect/process less features and data
  - Less complex models run faster
  - Models are easier to understand, verify and explain

Feature selection: definition

- Given a set of features $F = \{f_1, \ldots, f_i, \ldots, f_n\}$ the **Feature Selection problem** is to find a subset $F' \subseteq F$ that maximizes the learner ability to classify patterns.
- Formally F' should maximize some scoring function $\Theta : \Gamma \longrightarrow \mathbb{R}$ (where Γ is the space of all possible feature subsets of F), i.e.

$$\boxed{F' = \arg\max_{G \in \Gamma} \{\Theta(G)\}}$$

TÉCNICO LISBOA

Feature extraction: definition

- Given a set of features $F = \{f_1, \ldots, f_i, \ldots, f_n\}$ the **Feature Extraction (or Construction) problem** is to map F to some feature set F'' that maximizes the learner ability to classify patterns:

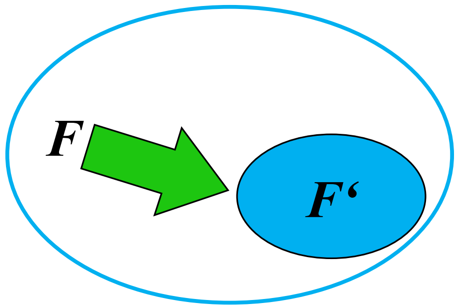
$$\boxed{F'' = \arg\max_{G \in \Gamma} \{\Theta(G)\}}$$

- This general definition subsumes feature selection (i.e. a feature selection algorithm also performs a mapping but can only map to subsets of the input variables)

TÉCNICO LISBOA

Feature selection vs. Feature extraction

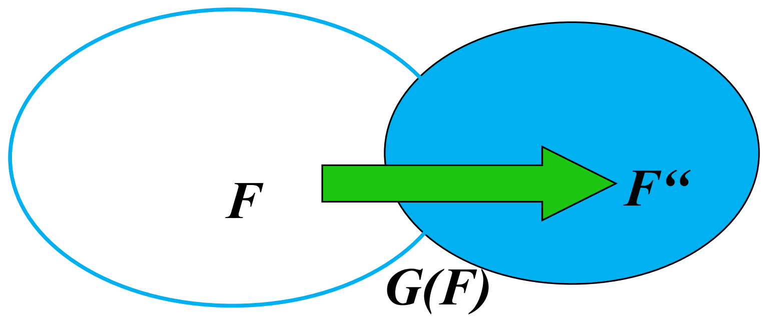
- **Feature Selection:**



$$\{f_1,\dots,f_i,\dots,f_n\} \xrightarrow[f.selection]{} \{f_{i_1},\dots,f_{i_j},\dots,f_{i_m}\}$$

$$i_j \in \{1,\dots,n\};\ j=1,\dots,m$$
$$i_a = i_b \Rightarrow a=b;\ \ a,b\in\{1,\dots,m\}$$

- **Feature Extraction/Creation:**



$$\{f_1,\dots,f_i,\dots,f_n\} \xrightarrow[f.extraction]{} \{g_1(f_1,\dots,f_n),\dots,g_j(f_1,\dots,f_n),\dots,g_m(f_1,\dots,f_n)\}$$

Feature selection: optimality

- In theory the **goal** is to find an **optimal feature-subset** (one that maximizes the scoring function).
- In real world applications this is usually **not possible**
  - For most problems it is computationally intractable to search the whole space of possible feature subsets.
  - One usually must settle for **approximations of the optimal subset**.
  - Most of the research in this area is devoted to finding efficient search heuristics.

Relevance of features

- **Relevance** vs **optimality** of feature set
  - Classifiers induced from training data are likely to be **suboptimal** (no access to the real distribution of the data).
  - Relevance **does not imply** that the feature is in the optimal feature subset.
  - Even "**irrelevant**" features **can improve** a classifier performance.
  - Defining **relevance in terms of a given classifier** (and therefore a hypothesis space) would be better.

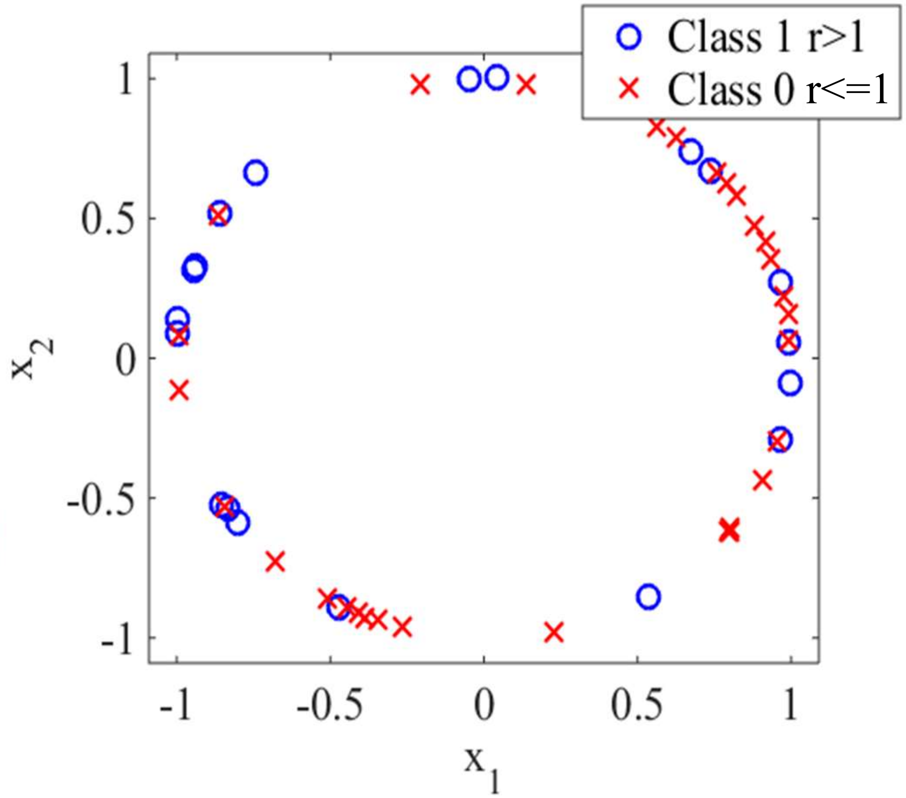
Test example

- Problem definition:

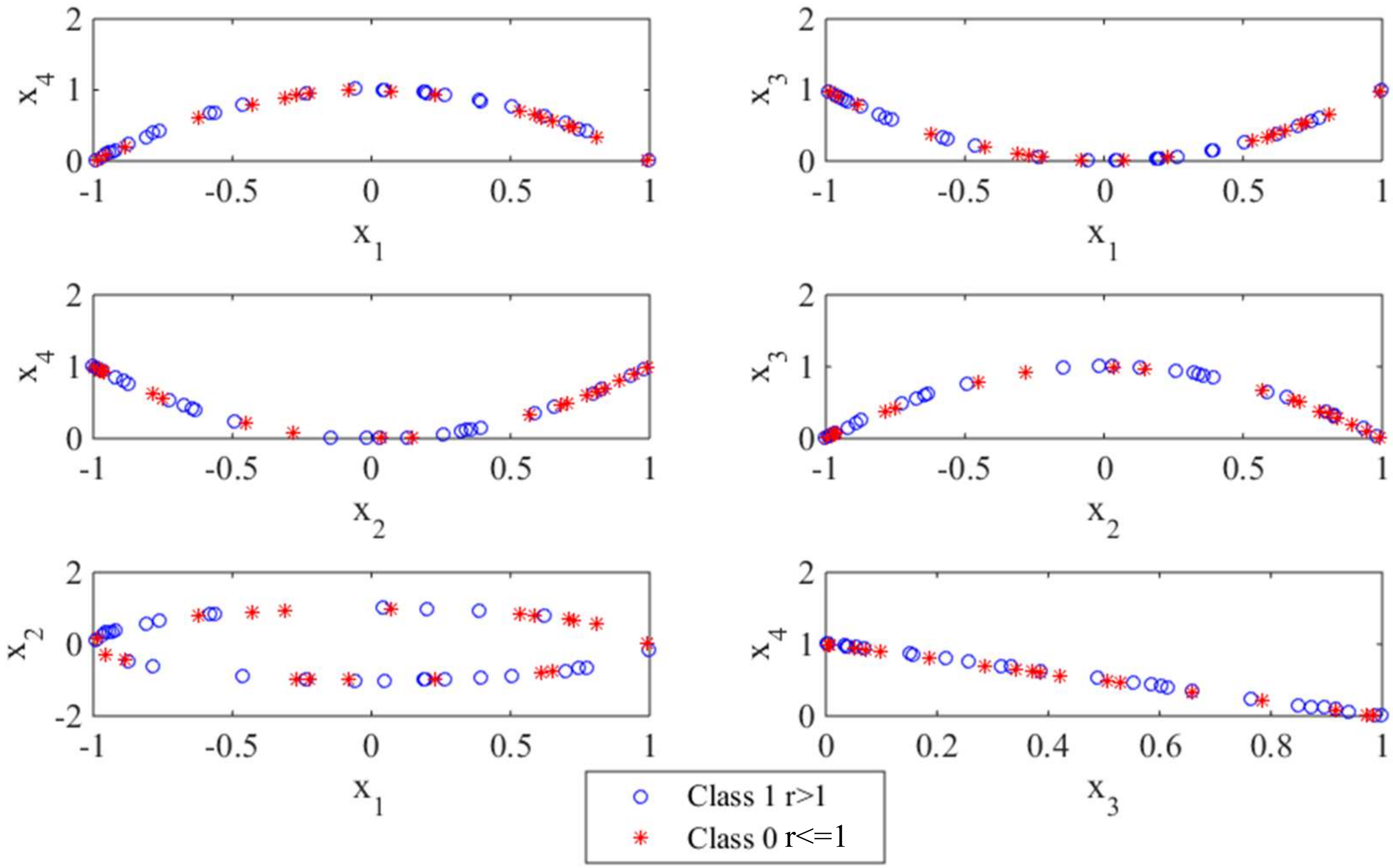
$$x_1 = r\cos(t)$$
$$x_2 = r\sin(t)$$
$$r \in [0.99, 1.01]$$
$$y = r > 1$$

- Features: $F = \begin{bmatrix} x_1 & x_2 & x_1^2 & x_2^2 \end{bmatrix}$

- Output: $y = \begin{bmatrix} 0 & 1 \end{bmatrix}$



Test example



Test example

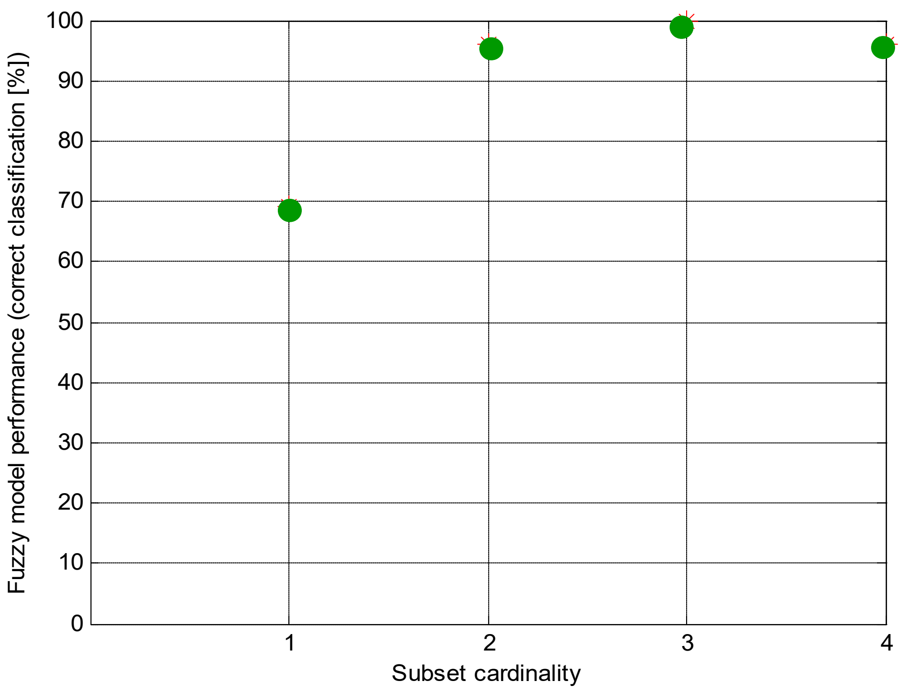
- Features: $F = \begin{bmatrix} x_1 & x_2 & x_1^2 & x_2^2 \end{bmatrix}$
- Output: $y = [0 \quad 1]$
- Correlation:

		x_1	x_2	x_3	x_4	
		x_1	x_2	x_1^2	x_2^2	y
x_1	x_1	1.0000	-0.1163	-0.1784	0.1790	-0.1090
x_2	x_2	-0.1163	1.0000	0.2002	-0.2085	-0.1162
x_3	x_1^2	-0.1784	0.2002	1.0000	-0.9995	0.1050
x_4	x_2^2	0.1790	-0.2085	-0.9995	1.0000	-0.0772
	y	-0.1090	-0.1162	0.1050	-0.0772	1.0000

TÉCNICO LISBOA

Test example

- All combinations of features using fuzzy models



Test example

- All possible combinations of feature subsets:
  - N(1) = {1}, {2}, {3}, {4}
  - N(2) = {1,2}, {1,3}, {1,4}, {2,3}, {2,4}, {3,4}
  - N(3) = {1,2,3}, {1,2,4}, {1,3,4}, {2,3,4}
  - N(4) = {1,2,3,4}
- Accuraccy for all combinations using fuzzy models:
  - N(1) = [46.1538] [50] [69.2308] [57.6923]
  - N(2) = [53.8462] [50] [53.8462] [50] [50] [96.1538]
  - N(3) = [53.8462] [53.8462] [100] [92.3077]
  - N(4) = [96.1538]

TÉCNICO LISBOA

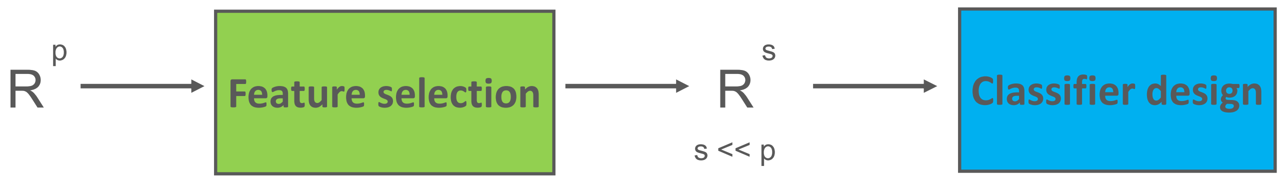
Feature selection

- **Filters**
  - Based on general characteristics of data to be evaluated.
  - No model is involved.
- **Wrappers**
  - Uses model performance to evaluate feature subsets.
  - Train one model for each feature subset.
- **Embedded methods**
  - Do not retrain the model at every step.
  - Search feature selection space and model parameter space simultaneously.

Hybrid methods

TÉCNICO LISBOA

Filter methods



- Features are scored independently, and the top ***s*** are used by the classifier.
- **Score:** correlation, mutual information, t-statistic, F-statistic, Fisher score, Gini Index, p-value, etc.

✓**Easy to interpret.**
✓**Usually fast.**

Feature ranking

- Given a set of features F

  Variable ranking is the process of ordering the features by the value of some scoring function $S : F \longrightarrow \mathbb{R}$ (which usually measures **feature-relevance**)

- Resulting set:
  a permutation of F: $F' = \{ f_{i_1}, \ldots, f_{i_j}, \ldots f_{i_n} \}$ with

$$S(f_{i_j}) \geq S(f_{i_{j+1}}); \quad j = 1, \ldots, n-1;$$

- The score $S(f_i)$ is computed from the training data, measuring some criteria of feature f_i.
- By convention a high score is indicative for a valuable (relevant) feature.

TÉCNICO LISBOA

Feature ranking

- A simple method for feature selection using variable ranking is to select the ***k* highest ranked features** according to S.
- This is usually **not optimal.**
- But often preferable to other, more complicated methods.
- **Computationally efficient**: only calculation and sorting of n scores.

Feature ranking

Questions:

- Can variables with small score be automatically discarded ?

NO

- Can a useless variable (i.e. one with a small score) be useful together with others?

YES

- Can two variables that are useless by themselves be useful together?

YES

Feature ranking

Take home messages:

- Correlation between variables and target is **not enough** to assess relevance.
- Correlation/covariance between pairs of variables has to be considered too.

  (potentially difficult, examples: **Joint Mutual Information, Relief**)
- **Diversity** of features – **which one to choose?**

TÉCNICO LISBOA

Filter methods

Problems:

- **Redundancy in selected features**: features are considered independently and not measured on the basis of whether they contribute with new information.
- **Interactions** among features generally can not be explicitly incorporated.
- **Classifier has no say in what features should be used**: some scores may be more appropriate in conjunction with some classifiers than others.

Sometimes used as a pre-processing step for other methods.

TÉCNICO LISBOA

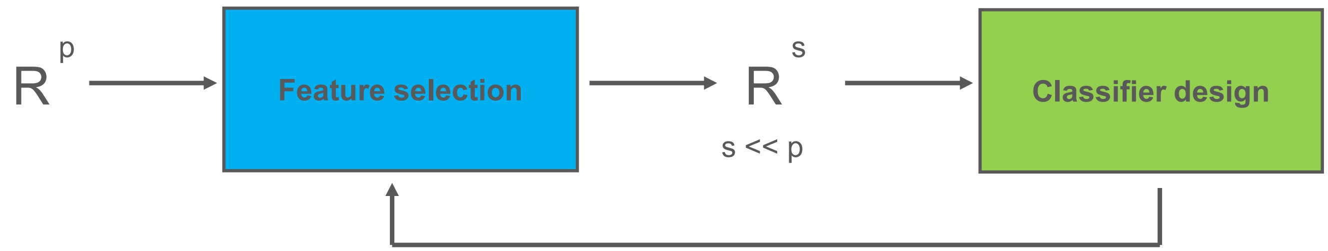
Dimension reduction

A variant of filter methods:

- Rather than retain a subset of ***s*** features, perform dimension reduction by projecting features onto ***s*** principal components of variation (e.g. PCA, etc.)
- Problem is that we are no longer dealing with one feature at a time but rather a linear or possibly more complicated combination of all features.

Those methods tend not to work better than simple filter methods and the model to build looses transparency.

Wrapper methods



- Iterative approach: many feature subsets are scored based on classification performance and best is used.
- **Selection of subsets**: forward selection, backward selection, forward-backward selection, ACO, GA, PSO, etc.

- **By using the learner as a black box, wrappers are universal.**

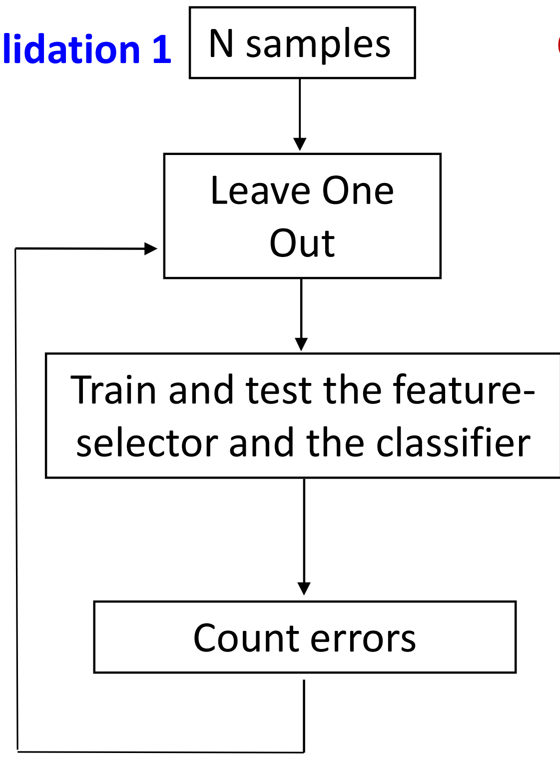
Wrapper methods

Problems:

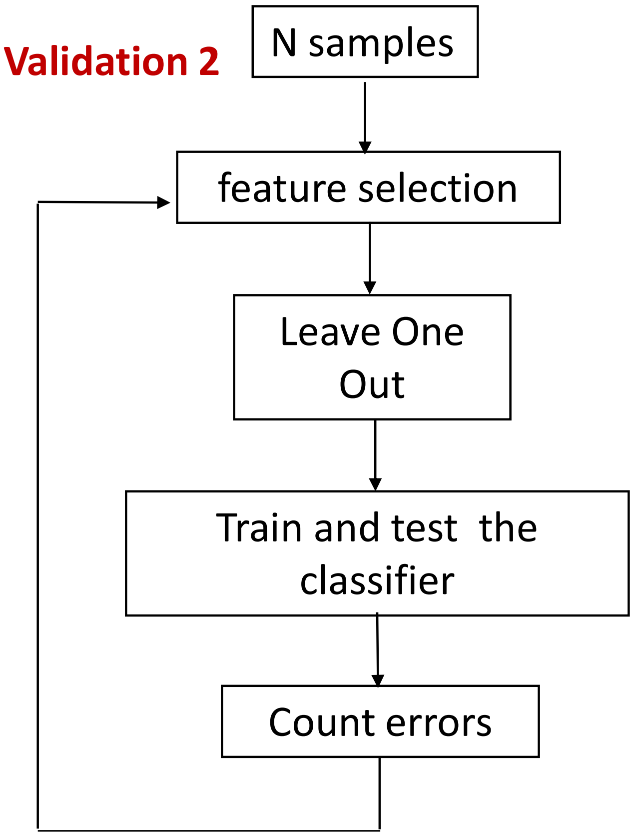
- **Computationally expensive**: for each feature subset to be considered, a classifier must be built and evaluated.
- No exhaustive search is possible (many subsets to consider): generally greedy algorithms only.
- **Easy to overfit**.

Validation

Cross Validation 1



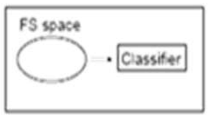
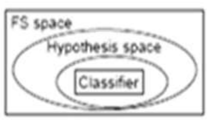
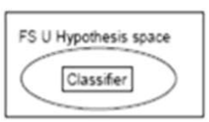
Cross Validation 2



CV2 – can yield optimistic estimation of classification true error.

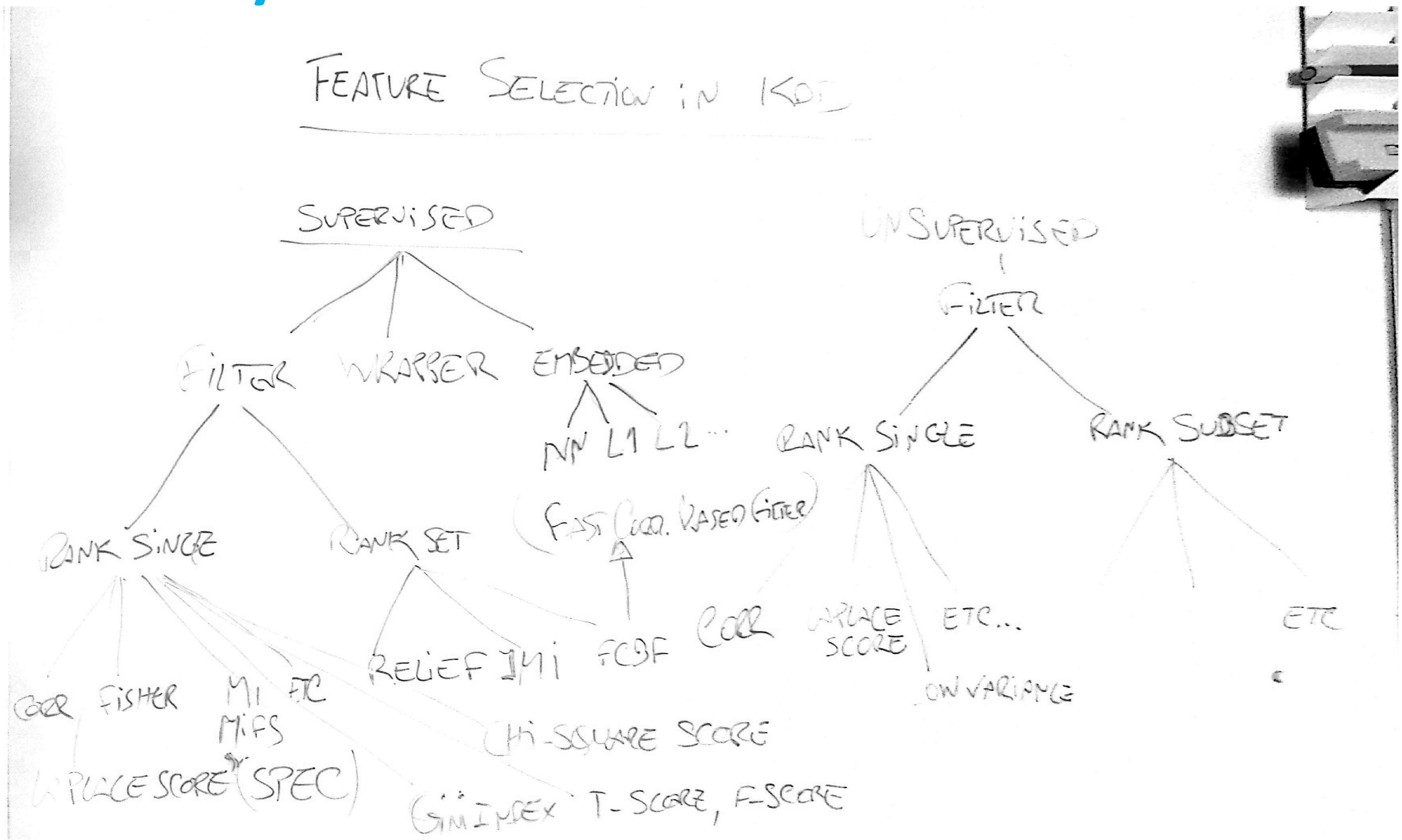
Taxonomy of feature selection

Table 1. A taxonomy of feature selection techniques. For each feature selection type, we highlight a set of characteristics which can guide the choice for a technique suited to the goals and resources of practitioners in the field.

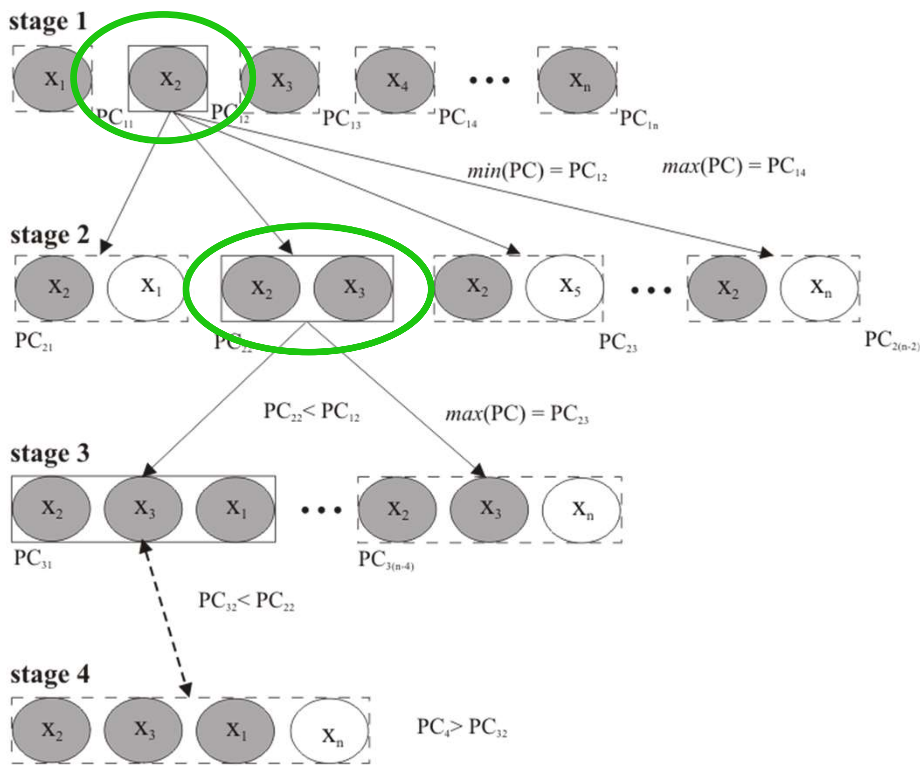
	Model search		Advantages	Disadvantages	Examples
Filter		Univariate	Fast Scalable Independent of the classifier	Ignores feature dependencies Ignores interaction with the classifier	Chi-square Euclidean distance t-test Information gain, Gain ratio [6]
		Multivariate	Models feature dependencies Independent of the classifier Better computational complexity than wrapper methods	Slower than univariate techniques Less scalable than univariate techniques Ignores interaction with the classifier	Correlation based feature selection (CFS) [45] Markov blanket filter (MBF) [62] Fast correlation based feature selection (FCBF) [136]
Wrapper		Deterministic	Simple Interacts with the classifier Models feature dependencies Less computationally intensive than randomized methods	Risk of over fitting More prone than randomized algorithms to getting stuck in a local optimum (greedy search) Classifier dependent selection	Sequential forward selection (SFS) [60] Sequential backward elimination (SBE) [60] Plus q take-away r [33] Beam search [106]
		Randomized	Less prone to local optima Interacts with the classifier Models feature dependencies	Computationally intensive Classifier dependent selection Higher risk of overfitting than deterministic algorithms	Simulated annealing Randomized hill climbing [110] Genetic algorithms [50] Estimation of distribution algorithms [52]
Embedded			Interacts with the classifier Better computational complexity than wrapper methods Models feature dependencies	Classifier dependent selection	Decision trees Weighted naive Bayes [28] Feature selection using the weight vector of SVM [44, 125]

Saeys Y, Inza I, Larrañaga P. A review of feature selection techniques in bioinformatics. Bioinformatics. 2007 1;23(19):2507-17

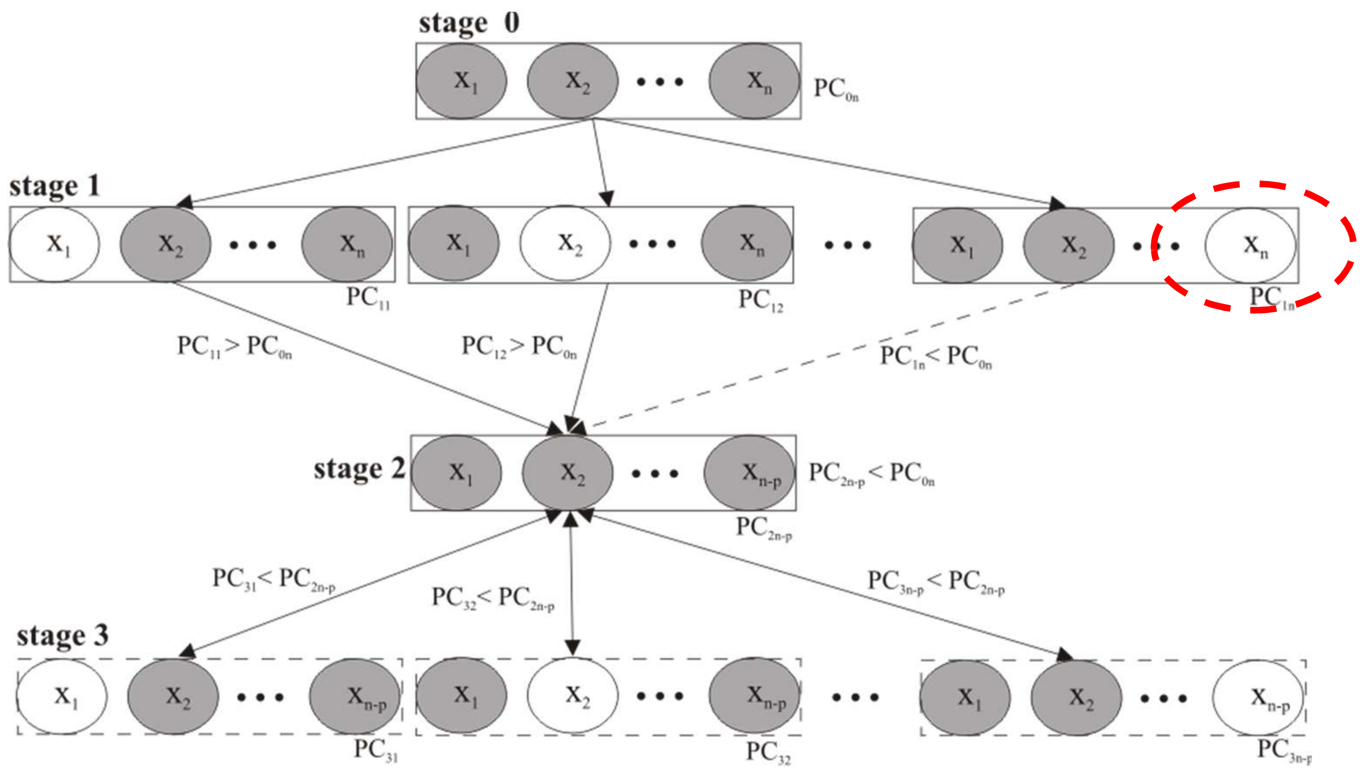
Taxonomy of feature selection



Tree search methods: SFS



Tree search methods: SBS



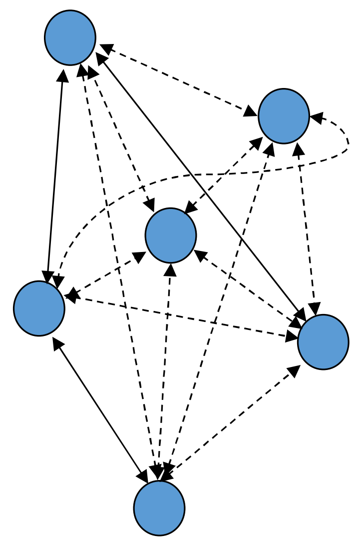
Tree search methods

- **Advantages:**
  - Easy to use
  - Reduce number of iterations (comparing to exhaustive search)
  - SFS achieves smaller number of features
- **Disadvantages:**
  - Converge to local minima
  - Computationally very heavy for more than about 50 features

Metaheuristic methods → global search

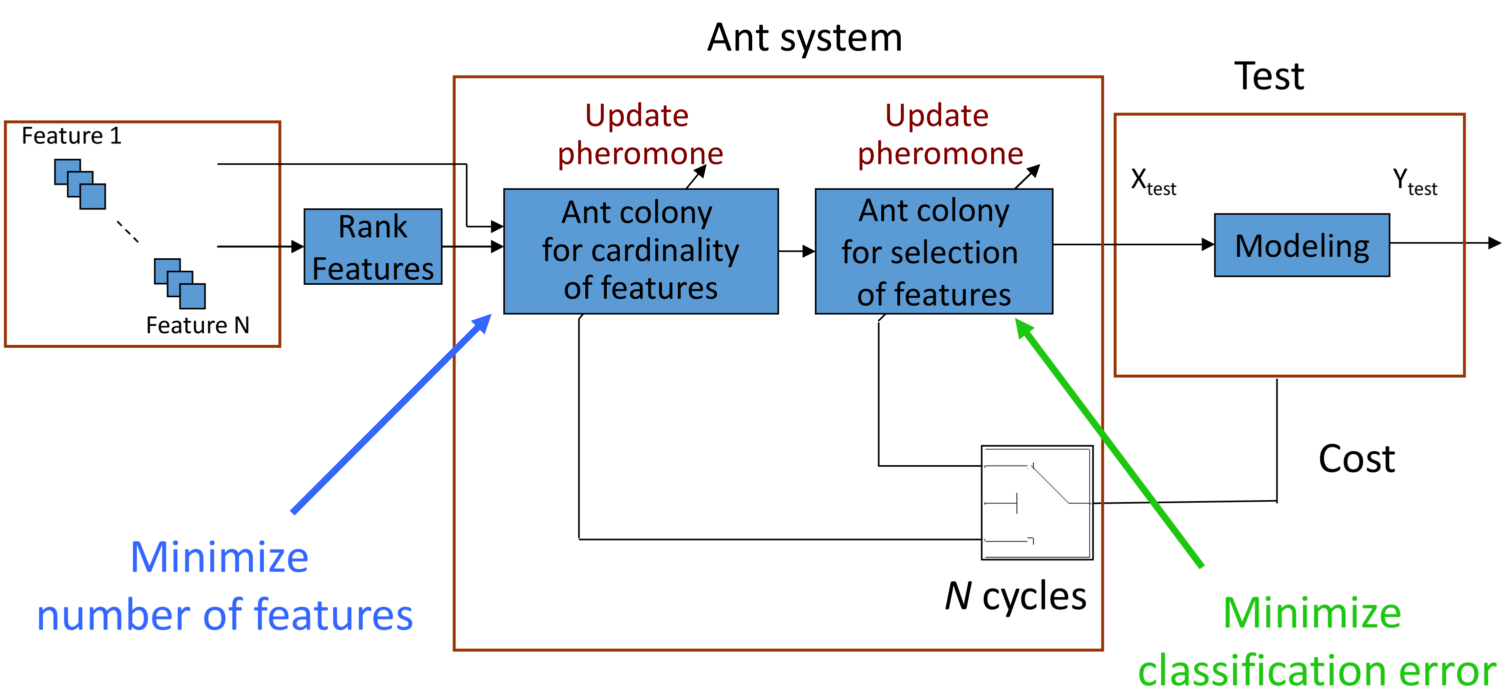
Artificial ants

- Artificial ants move in **graphs**
  - nodes / arcs
  - environment is discrete
- As real ants:
  - choose paths based on pheromone concentration
  - deposit pheromones on paths
  - environment updates pheromones
- Extra abilities of artificial ants:
  - prior knowledge (**heuristic** η)
  - memory (**feasible neighbourhood** N)



Ant feature selection

- Multicriteria algorithm (S. Vieira et al., 2010):



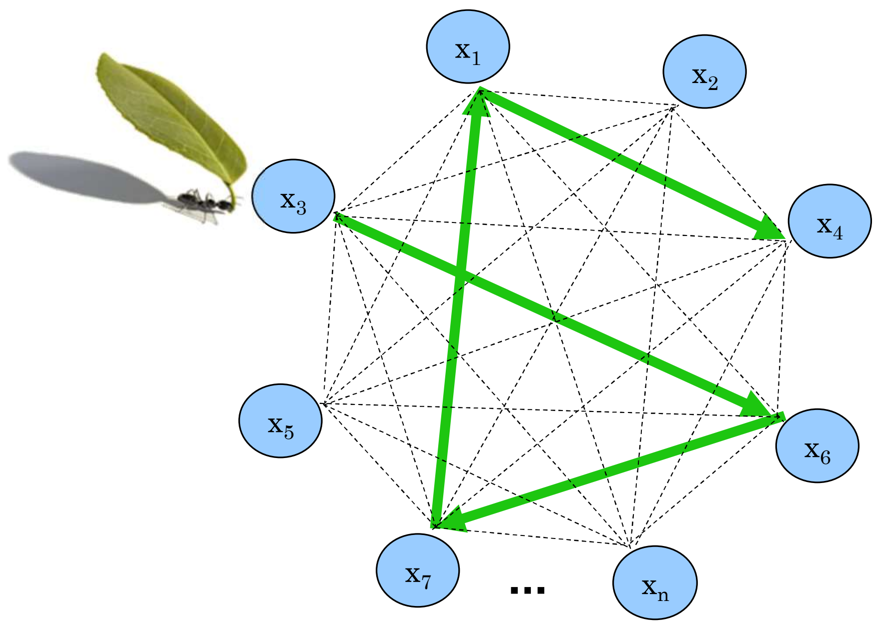
Ant feature selection

- **Choose node**

$$p_{ij}^{k} = \begin{cases} \dfrac{\tau_{ij}^{\ \alpha} \times \eta_{ij}^{\ \beta}}{\sum_{j \in \mathrm{N}} \tau_{ij}^{\ \alpha} \times \eta_{ij}^{\ \beta}}, & if \quad j \in N \\ 0, & \text{otherwise} \end{cases}$$

- **Pheromone update**

$$\tau(l+1) = \tau(l)(1-\rho) + \Delta\tau_{ij}^{k}$$



Subset:
$\{x_3, x_6, x_7, x_1, x_4\}$

Heuristics in AFS

- **Heuristic for feature cardinality**: Fisher's score for the features

$$F(i) = \frac{\left|\mu_{c_1}(i) - \mu_{c_2}(i)\right|^2}{\sigma_{c_1}^2(i) + \sigma_{c_2}^2(i)}$$

mean and variance values of feature i for the samples in class c_1 and c_2

- **Heuristic for selection of features**: classification error $e(i)$ for the individual features

$$\eta_f(i) = \frac{1}{e(i)}$$

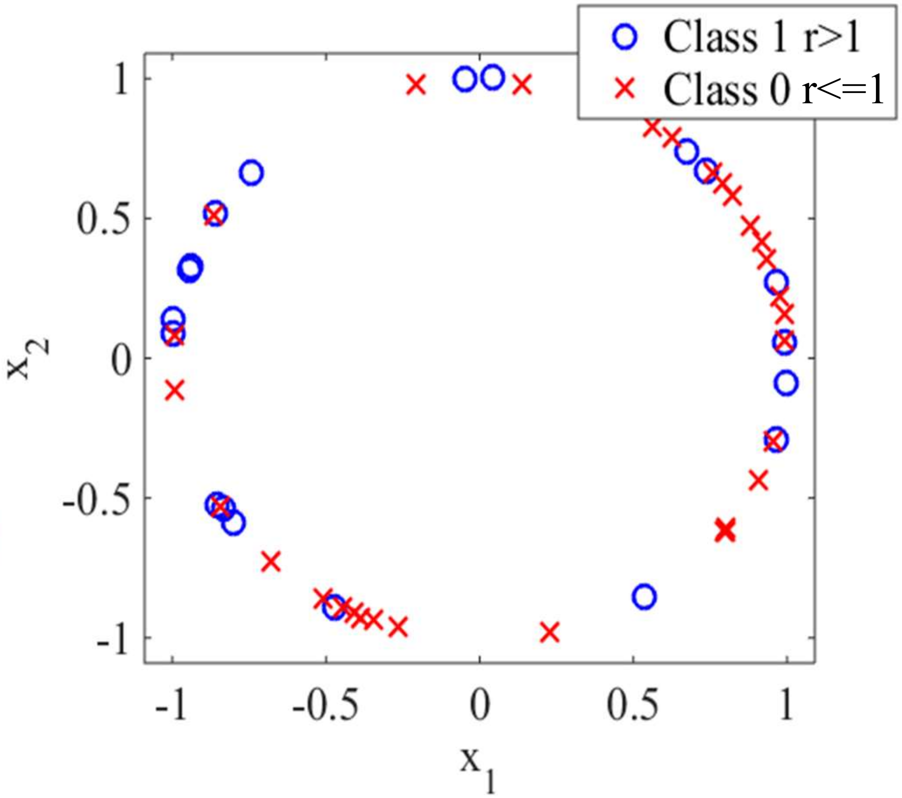
Test example

- Problem definition:

$$x_1 = r\cos(t)$$
$$x_2 = r\sin(t)$$
$$r \in [0.99, 1.01]$$
$$y = r > 1$$

- Features: $F = \begin{bmatrix} x_1 & x_2 & x_1^2 & x_2^2 \end{bmatrix}$

- Output: $y = [0 \quad 1]$

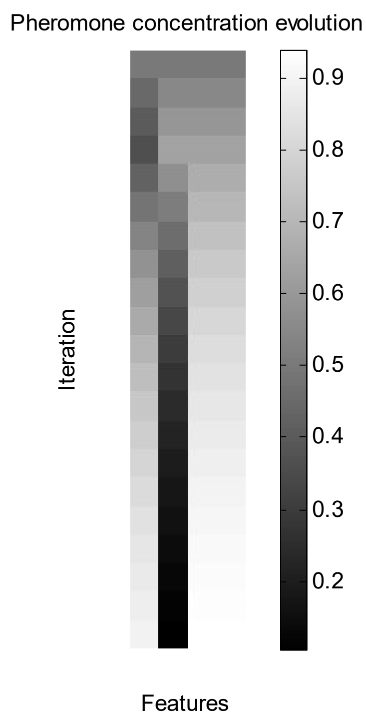


Test example

- All possible combinations of feature subsets:
  - N(1) = {1}, {2}, {3}, {4}
  - N(2) = {1,2}, {1,3}, {1,4}, {2,3}, {2,4}, {3,4}
  - N(3) = {1,2,3}, {1,2,4}, {1,3,4}, {2,3,4}
  - N(4) = {1,2,3,4}
- Accuraccy for all combinations using fuzzy models:
  - N(1) = [46.1538] [50] [69.2308] [57.6923]
  - N(2) = [53.8462] [50] [53.8462] [50] [50] [96.1538]
  - N(3) = [53.8462] [53.8462] [100] [92.3077]
  - N(4) = [96.1538]

Test example

- Ant feature selection using fuzzy models (5 ants, 20 iterations).



Results: fuzzy models

- Classification rates with 10-fold cross validation:

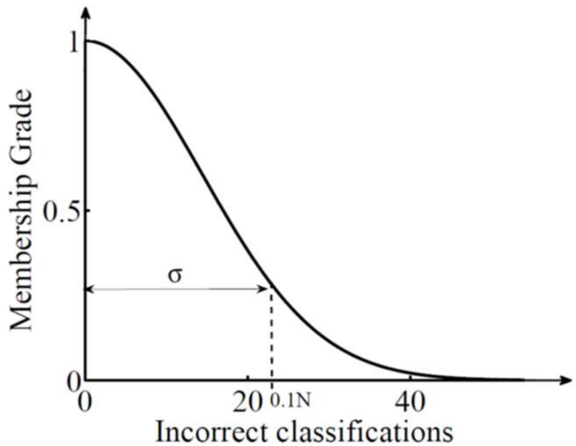
Data set	Fuzzy Models					
	Classification Accuracy		Standard deviation		Number of features	
	No FS	AFS	No FS	AFS	No FS	AFS
1 WBCO	84.5	97.7	1.75	1.21	9	2-5
2 Wine	82.6	99.5	3.40	1.66	13	2-4
3 Vote	80.0	99.7	4.18	1.02	16	2-5
4 WDBC	77.2	99.5	3.05	0.84	32	2-3
5 WPBC	78.9	85.6	1.50	2.47	33	2
6 Sonar	60.2	86.6	5.73	2.83	60	2-3
7 Musk	77.7	78.3	4.14	4.39	166	2-20
Average	**77.3**	**92.4**	-	-	-	-
WTL	0/0/7	0/1/6	-	-	-	-

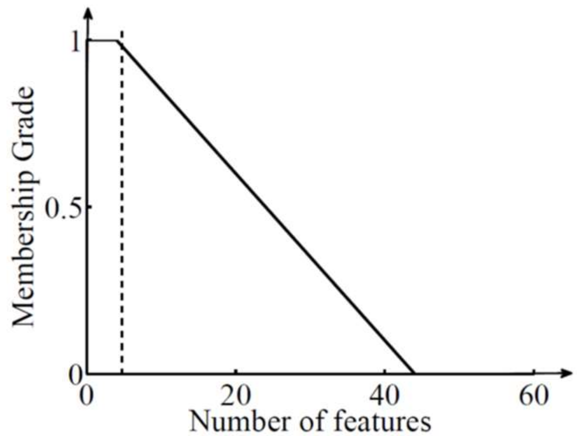
Fuzzy objective function

- **Classic objective function** $\text{minimize } f = w_1 e + w_2 N$

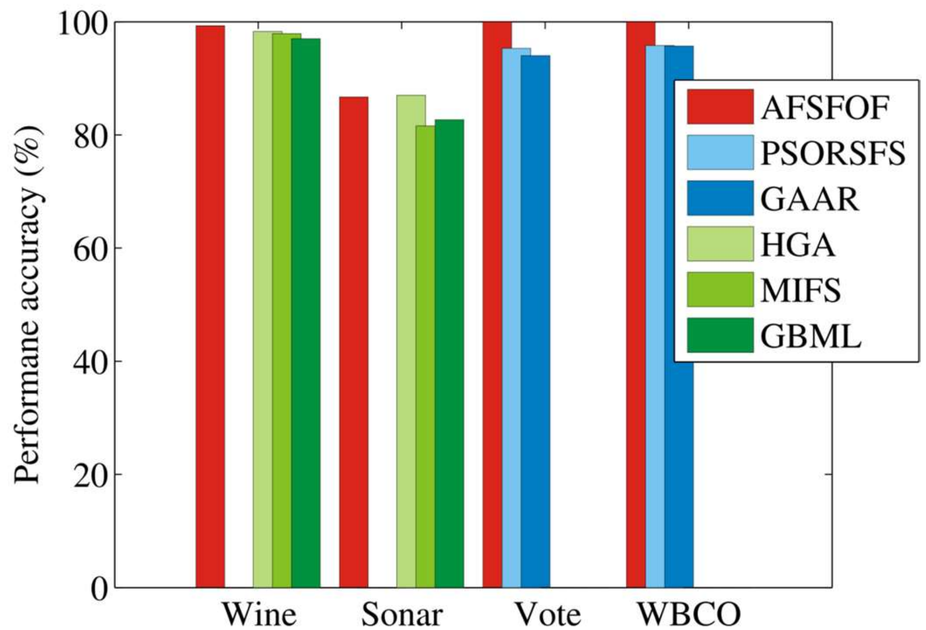
- **Fuzzy objective function**

$$\text{maximize } D(\mathbf{x})$$
$$D(\mathbf{x}) = \odot\, (I(F_1, w_1), I(F_2, w_2))$$





Comparison with state-of-the-art



GAAR - genetic algorithm-based
PSORSFS - particle swarm optimization algorithm-based
GBML – multi-objective fuzzy genetics-based machine learning
MIFS - a classical filter method based on mutual information
HGA - a hybrid genetic algorithm wrapper approach based on mutual information

Real world example

- **MEDAN database**
  - **Variables:**

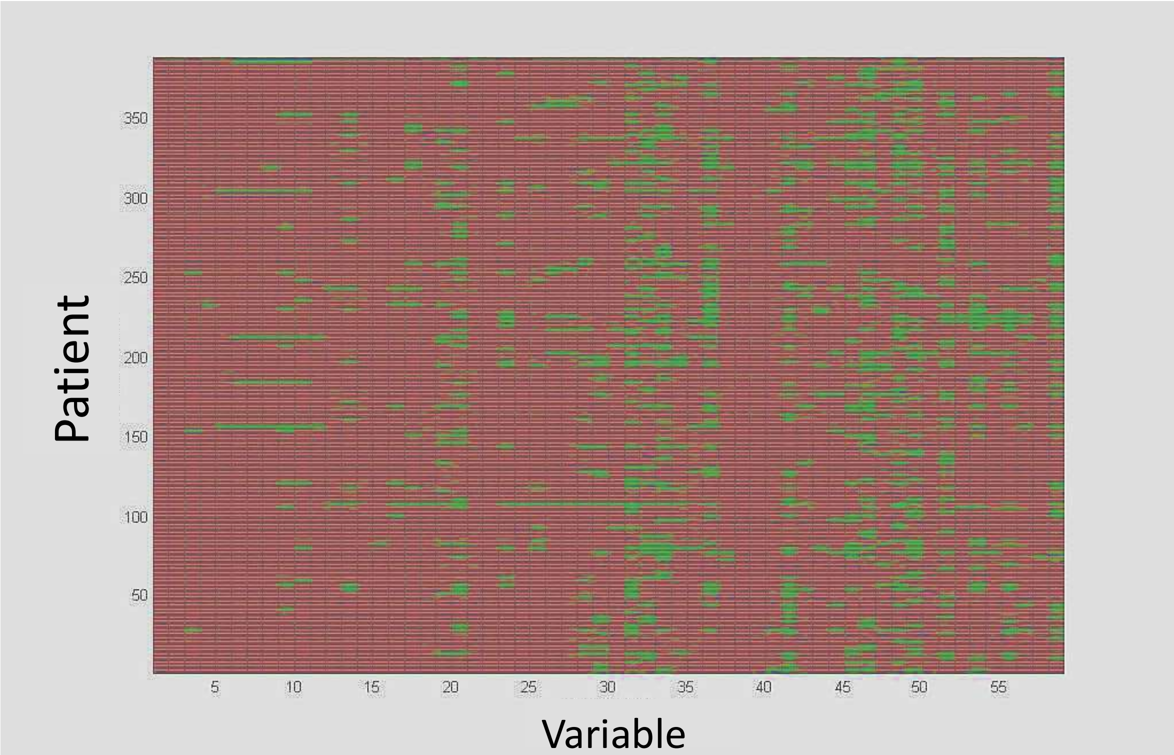
    The MEDAN data base contains the data of 382 patients. The data were copied from intensive care unit records in the years 1998-2002 by medical documentation staff. All patients have septic shock of abdominal cause.

  - **Task:**

    Predict patients survival.

  - Problems in the database…

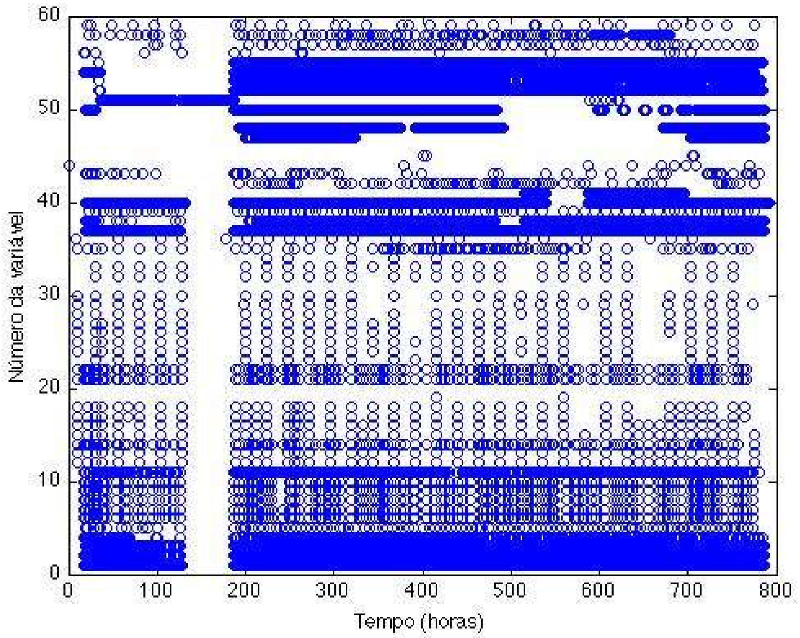
Sepsis patients database



The matrix contains 387 patients and 59 variables.

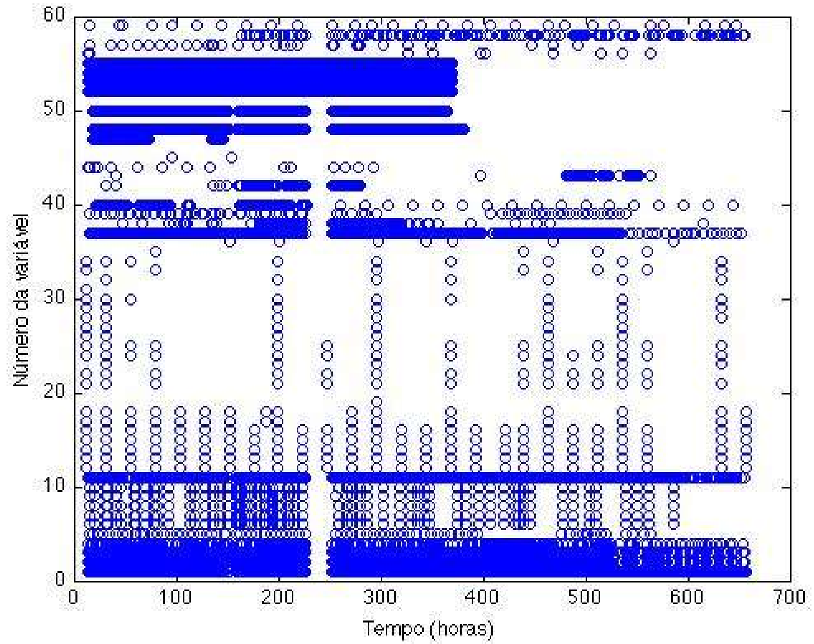
MEDAN - Problems

- Different time samples:



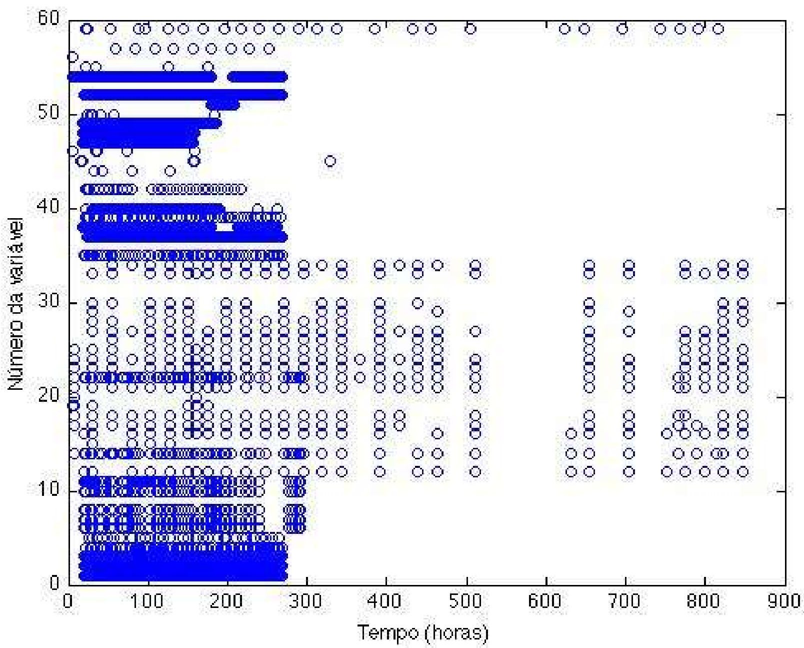
MEDAN - Problems

- Missing data:

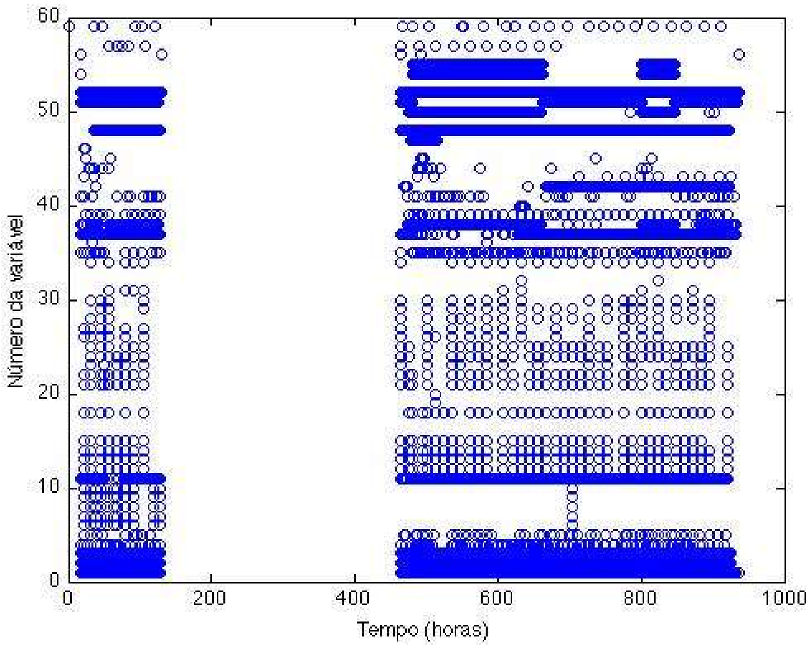


MEDAN - Problems

- Stopped being measured:



MEDAN - Problems

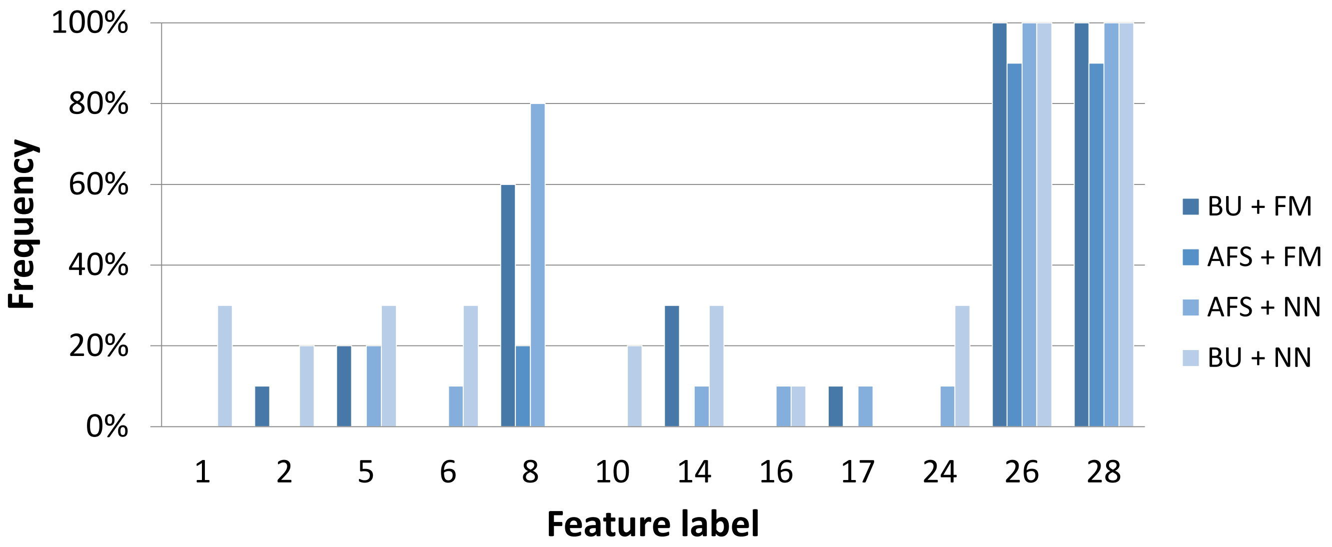


Classification accuracy (%)

- Results

FS method	Model	12 Features set			28 Features set		
		Num. Feat.	Mean	Std	Num. Feat.	Mean	Std
-	NN [Paetz]	12	69.0	**4.37**	-	-	-
Bottom -up	Fuzzy TS	2-6	74.1	1.31	2-7	**82.3**	1.56
	NN	2-8	73.2	2.03	4-8	81.2	1.97
AFS	Fuzzy TS	**2-3**	72.8	1.44	3-9	78.6	1.44
	NN	2-7	**75.7**	1.37	5-12	81.9	2.12

12 features subset



Most frequent features:

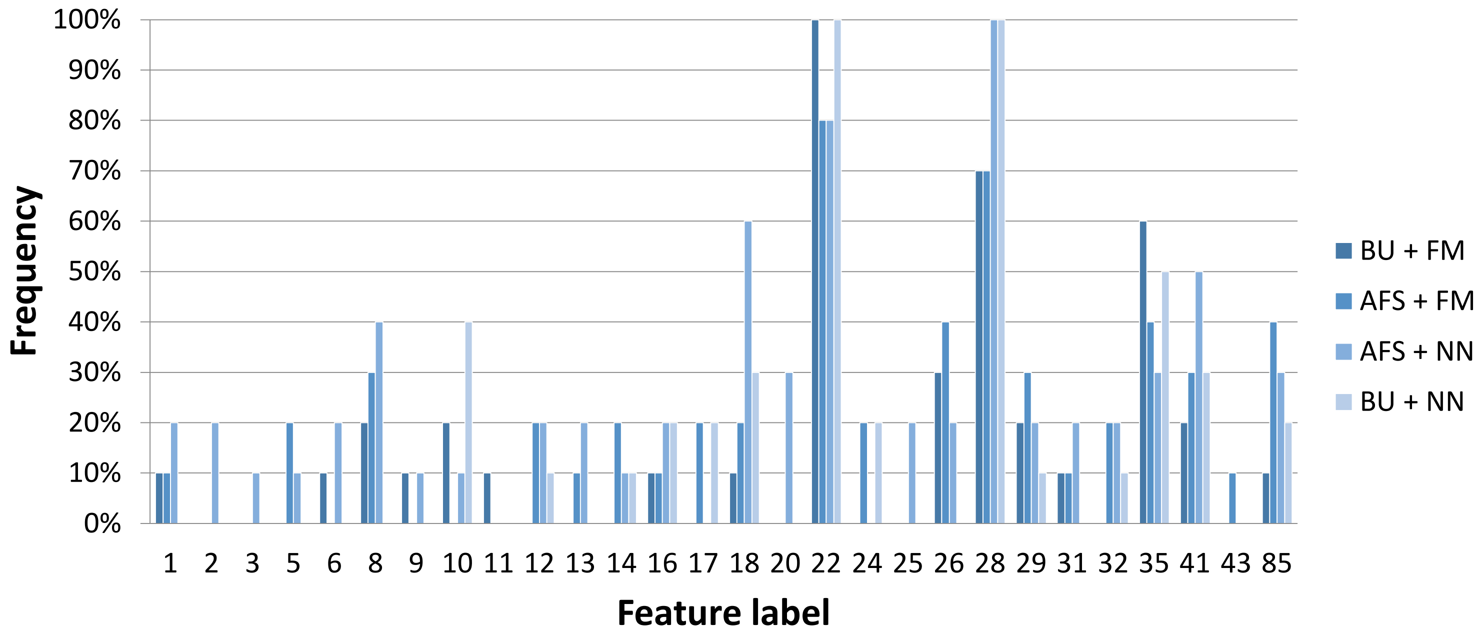
8 – pH

26 – Calcium

28 – Creatinine

TÉCNICO LISBOA

28 features subset



Most frequent features (besides previous 8, 26 and 28):

18 – thrombocytes
22 – antithrombin III
35 – total bilirubin
41 – CRP (C-reactive protein)
85 – FiO2

TÉCNICO LISBOA