

## Week 8

# Course. Introduction to Machine Learning

## Theory 8. Feature Selection

Dr. Maria Salamó Llorente

[maria.salamo@ub.edu](mailto:maria.salamo@ub.edu)

Dept. Mathematics and Informatics,  
Faculty of Mathematics and Informatics,  
University of Barcelona (UB)

# Introduction to Machine Learning

## Unsupervised Learning

Cluster Analysis

Factor Analysis

Visualization

K-Means,  
Fuzzy C-means,  
EM

PCA, ICA

Self Organized Maps (SOM) ,  
Multi-Dimensional Scaling

Lazy Learning  
(K-NN, IBL,  
CBR)

**Overfitting,  
model selection  
and feature selection**

Kernel Learning

Ensemble Learning  
(Trees, Adaboost )

Perceptron,  
SVM

## Supervised Learning

Non Linear Decision

Linear Decision

Decision Learning Theory

Basic concepts of  
Decision Learning Theory

1. Introduction to feature selection
2. Feature selection perspectives
3. Aspects in feature selection:
  1. Output of the feature selection
  2. Evaluation
  3. Drawbacks
4. Most representative feature selection methods
5. Summary



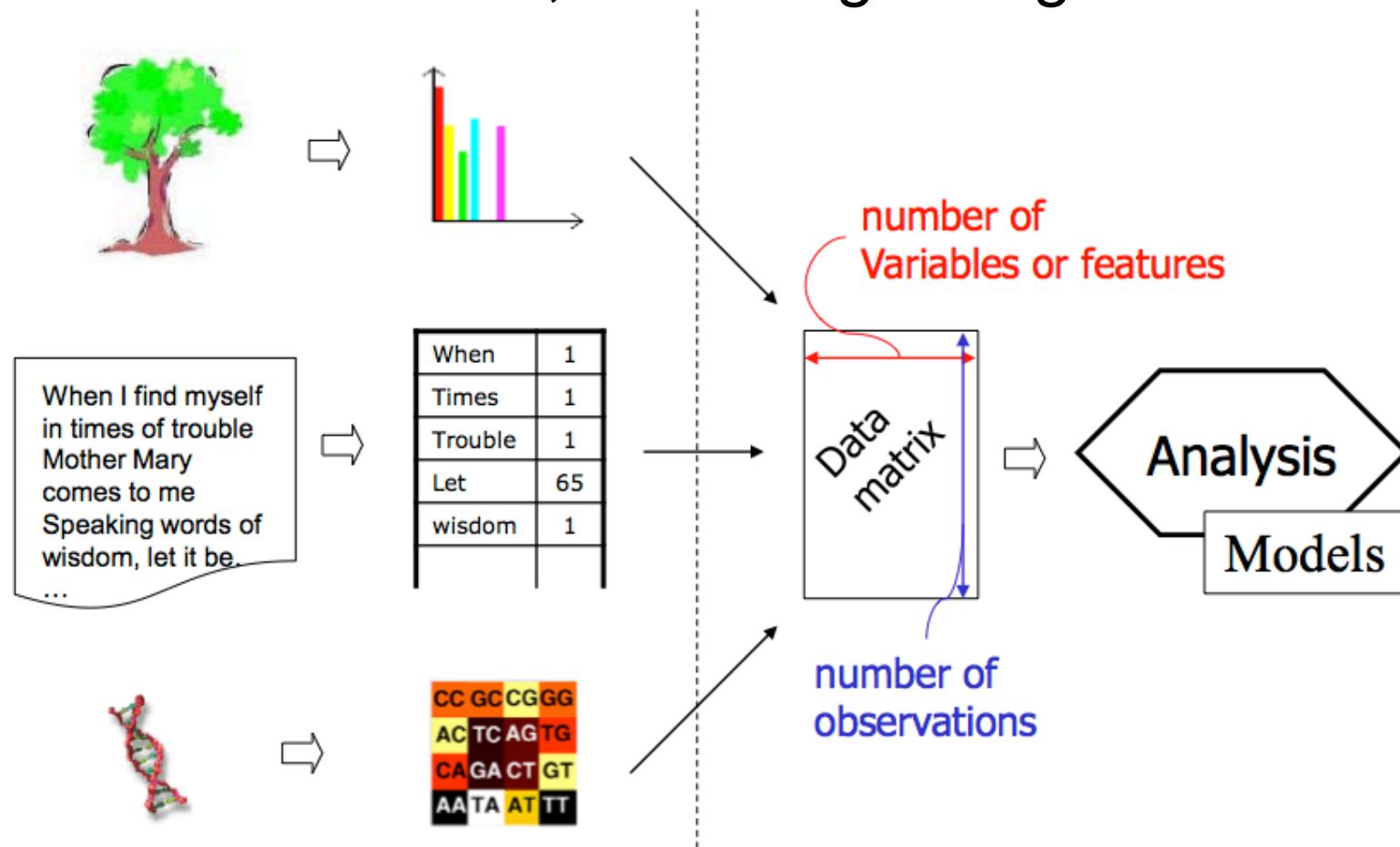
# Introduction to feature selection

## Problem: Where to focus attention?

- A universal problem of intelligent (learning) systems is where to focus their attention
- What aspects of the problem at hand are important/necessary to solve it?
- Discriminate between the relevant and irrelevant parts of experience

# Introduction to feature selection

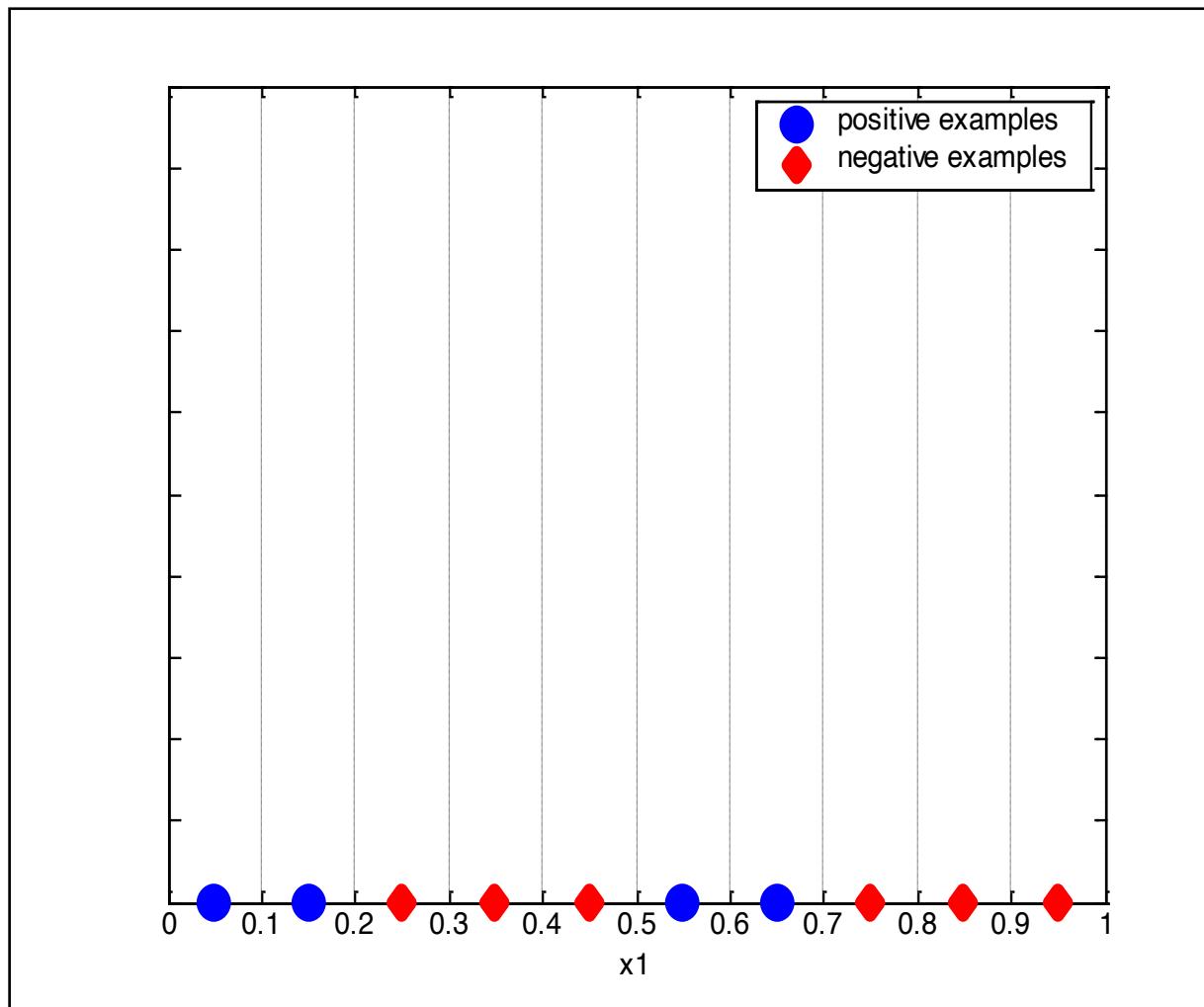
- High-dimensional data exists
  - For classification, clustering or regression



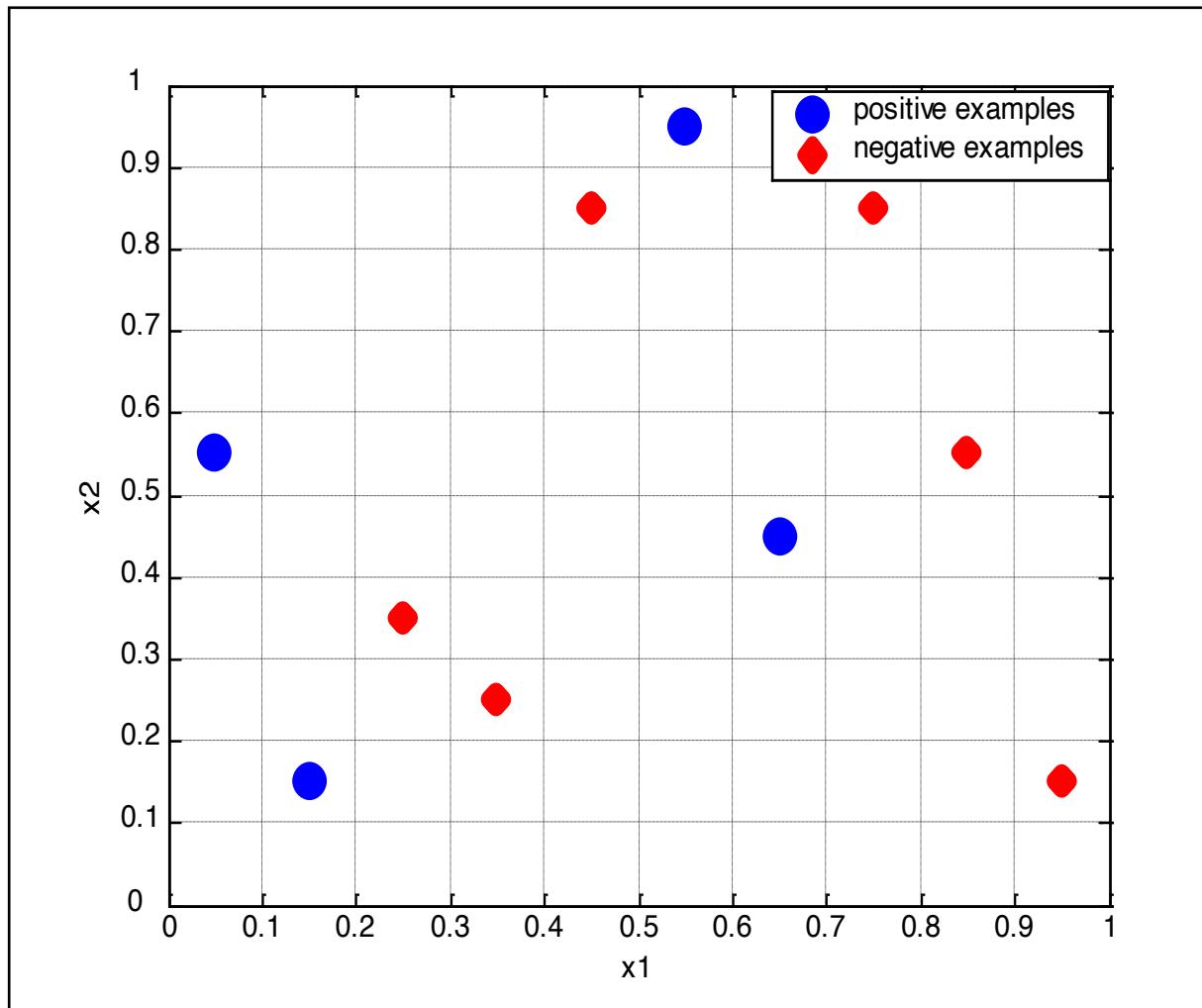
## Why reducing dimensionality?

- Theoretically not useful:
  - More information means easier task
  - Models can ignore irrelevant features
    - “In theory, practice and theory are the same. But in practice, they are not”
- Lots of inputs means ...
  - Lots of parameters
  - Large input space
  - **Curse of dimensionality** and risks of **overfitting!**

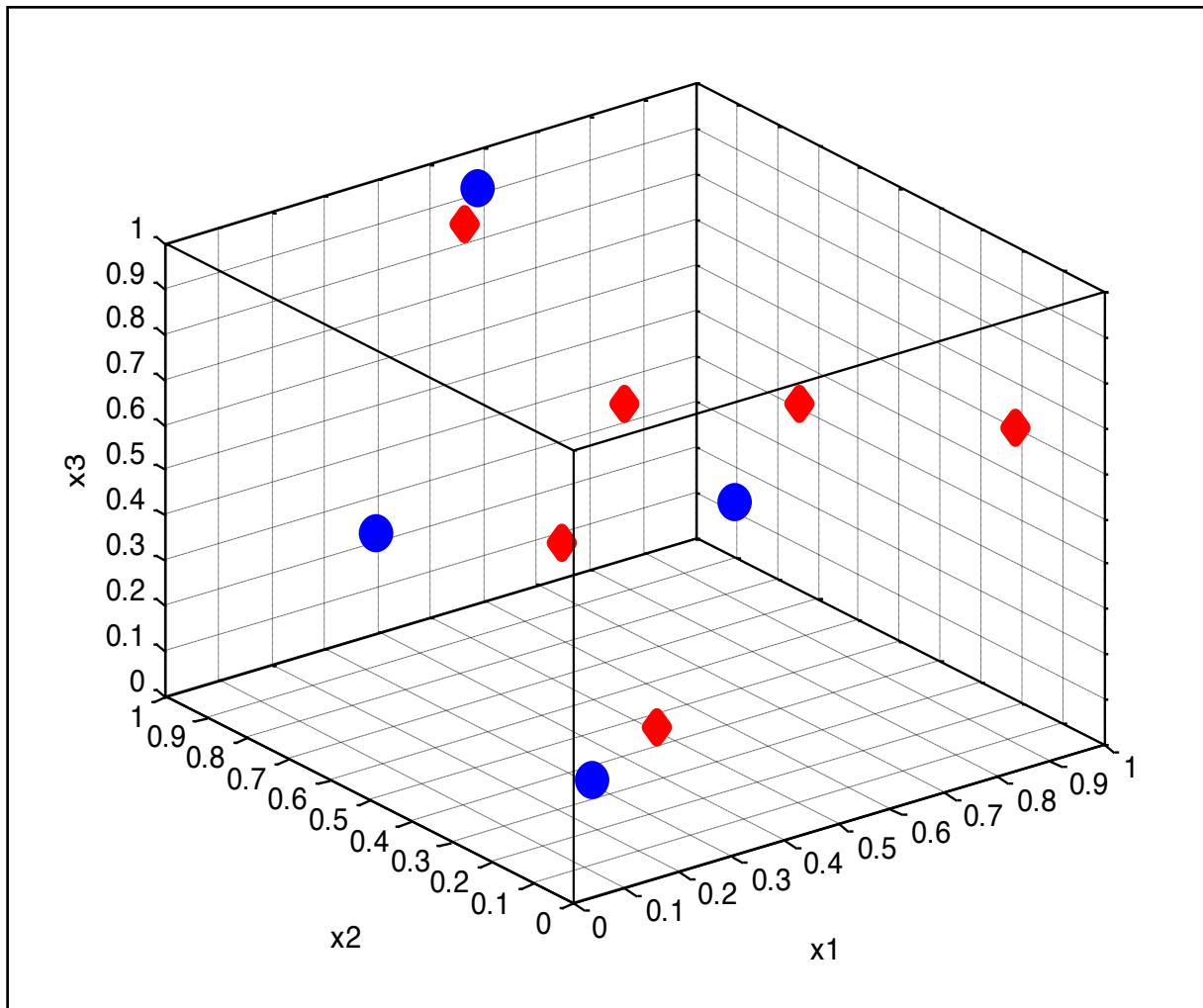
# Curse of dimensionality



# Curse of dimensionality

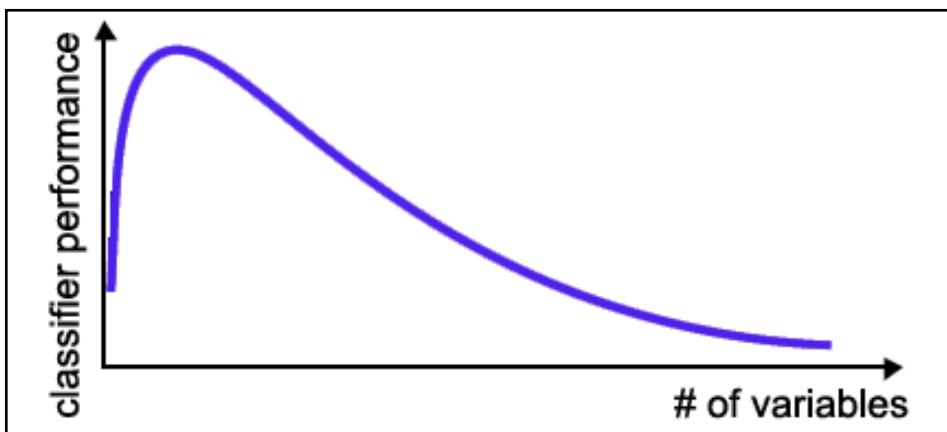


# Curse of dimensionality



# Curse of Dimensionality

- 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!
  - The classifier's performance usually will degrade for a large number of features!



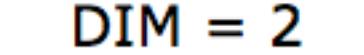
*In many cases the information that is lost by discarding variables is made up for by a more accurate mapping/sampling in the lower-dimensional space !*

# Curse of dimensionality

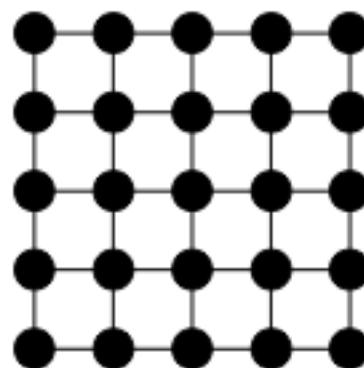
- The number of points in a grid increases exponentially with dimension (DIM)



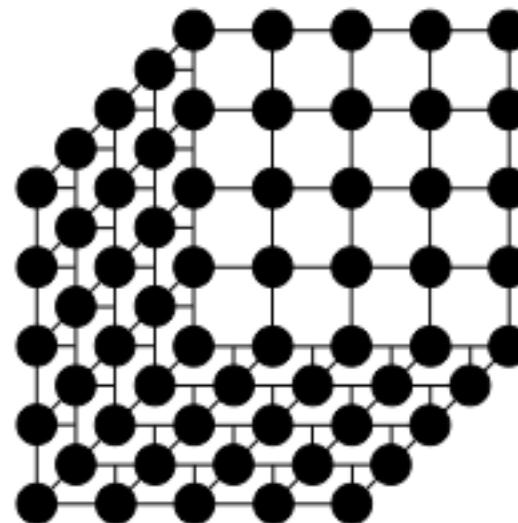
DIM = 1



DIM = 2



DIM = 3



- In high DIM:
  - Never enough data
  - Never sure to interpolate



***Feature selection  
reduces dimensionality***

# Introduction to feature selection

- **Definition:** Feature selection is a process that chooses an optimal subset of features according to a certain criterion [Liu and Motoda, 1998]
- *Feature selection* is typically a **search problem** for finding an optimal or suboptimal subset of  $m$  features out of original  $M$  features.
  - For large number of features, exhaustive search for the best subset out of  $2^M$  possible subsets is infeasible
- **Benefits of feature selection:**
  - For excluding irrelevant and redundant features,
  - it allows reducing system complexity and processing time,
  - and often improves the prediction accuracy.

- **Why we need FS:**

1. To **improve performance** (in terms of speed, predictive power, simplicity of the model)
2. To **visualize the data** for model selection
3. To **reduce dimensionality** and **remove noise**
4. To **reduce the cost of the data**
5. To **reduce the complexity** of the resulting model description

## *Unsupervised*

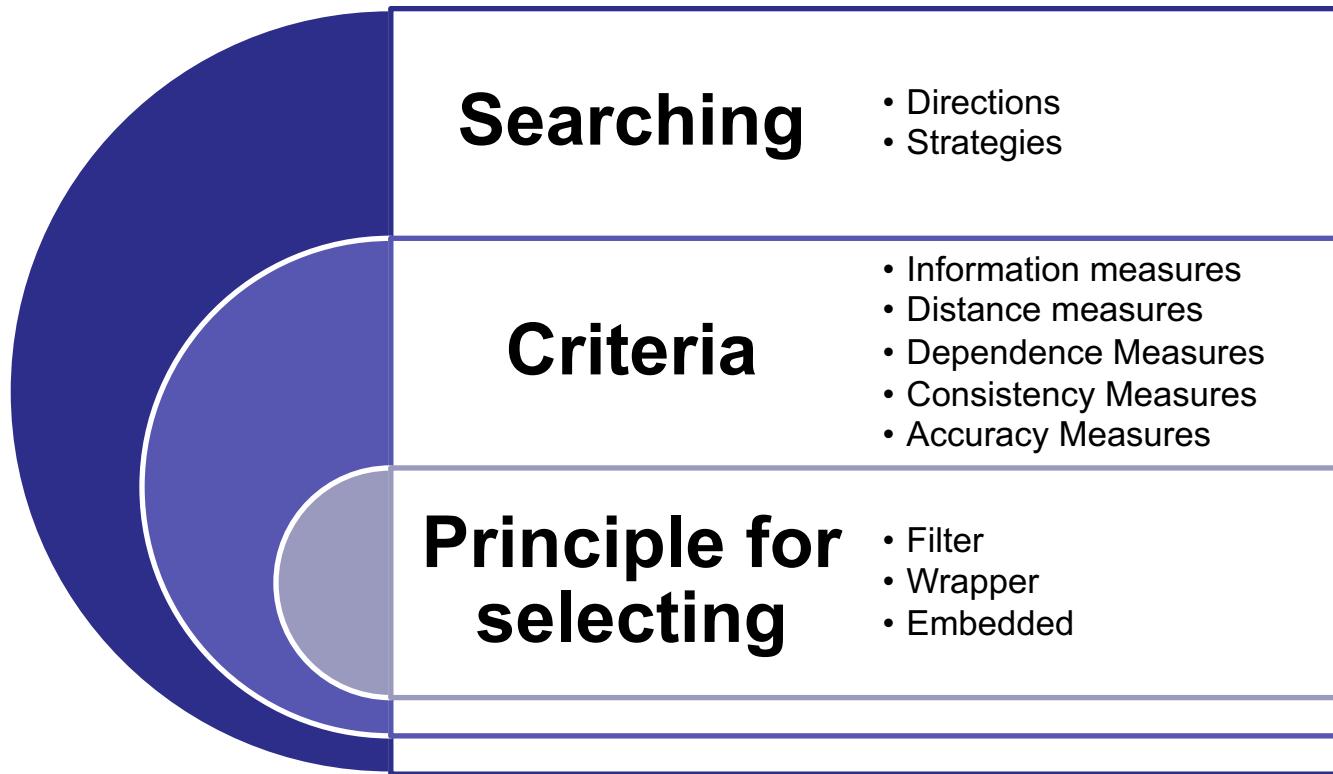
	<b>Linear</b>	<b>Nonlinear</b>
<b>Selection</b>	Correlation between inputs	Mutual information between inputs
<b>Projection</b>	Principal Component Analysis	Sammon's Mapping Self-Organized Maps

## *Supervised*

	<b>Linear</b>	<b>Nonlinear</b>
<b>Selection</b>	Correlation between inputs and outputs	Mutual information between inputs and outputs, Greedy algorithms, Genetic algorithms
<b>Projection</b>	Linear Discriminant Analysis, Partial Least Squares	Projection pursuit

# Feature selection perspectives

# Feature selection perspectives



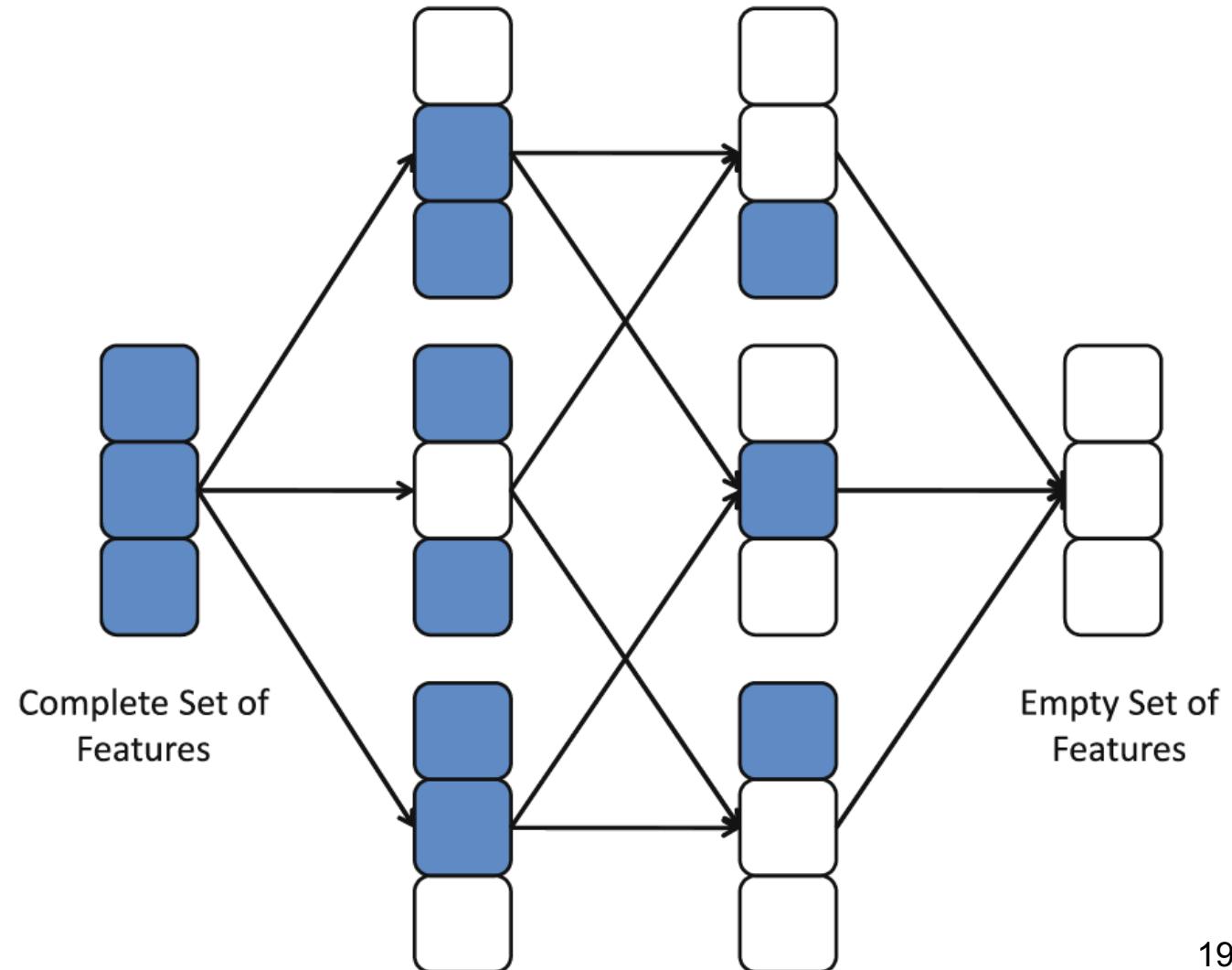
1. **searching** for the best subset of features
2. **criteria** for evaluating different subsets
3. **principle for selecting**, adding, removing or changing new features during the search

## Search of a Subset of Features

- FS can be considered as a **search problem**, where each state of the search space corresponds to a concrete subset of features selected.
- The selection can be represented as a binary array, with each element corresponding to the value 1, if the feature is currently selected by the algorithm and 0, if it does not occur.
- There should be a total of  $2^M$  subsets where  $M$  is the number of features of a data set.

## Search of a Subset of Features

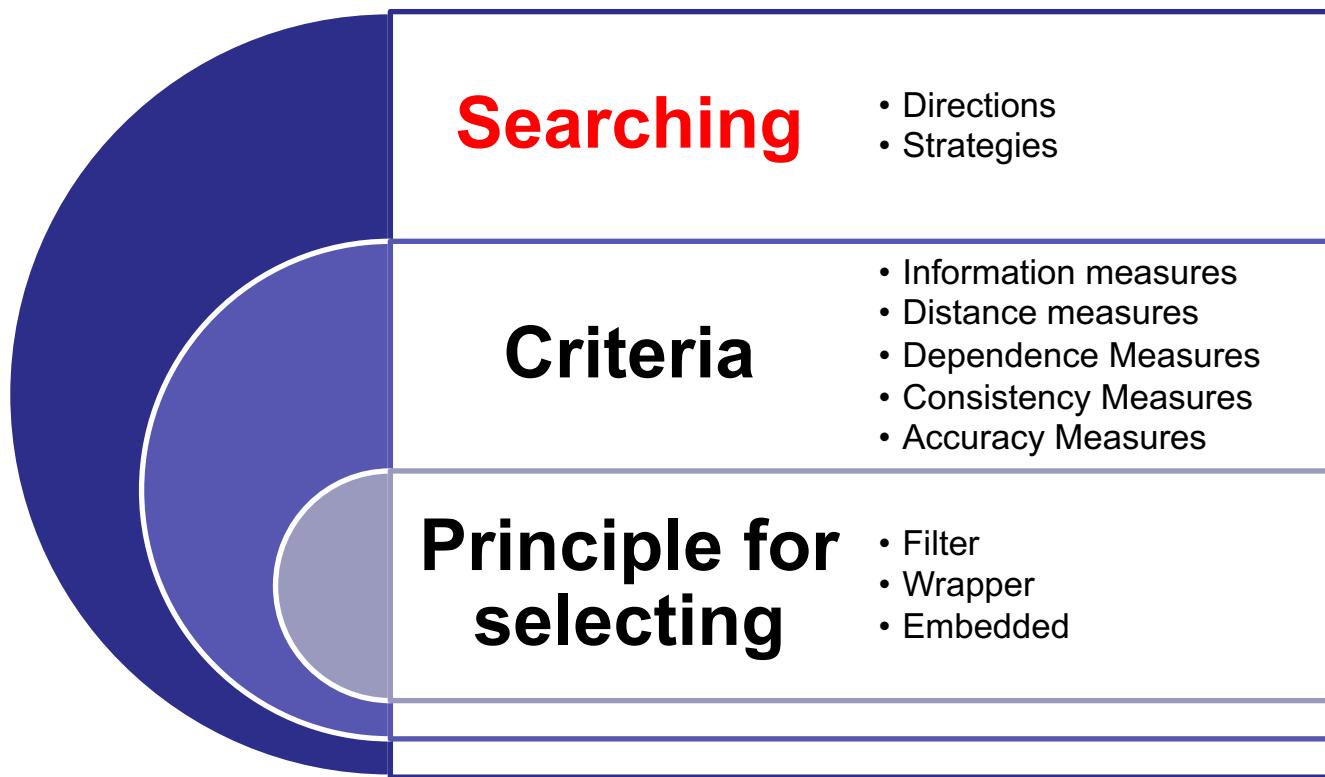
*Search  
Space:*



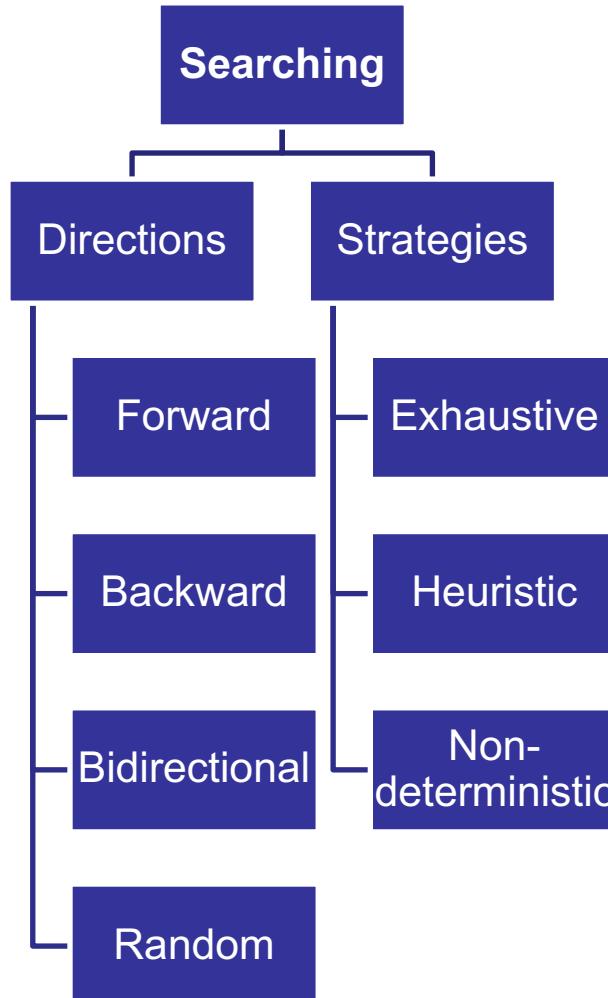
# Search Strategies

- Assuming  $n$  features, an exhaustive search would require:
  - Examining all  $\binom{n}{d}$  possible subsets of size  $d$ .
  - Selecting the subset that performs the best according to the criterion function.
- The number of subsets grows in a **combinatorial** way, making exhaustive search impractical
- In practice, heuristics are used to speed-up search but they **cannot** guarantee optimality

# Feature selection perspectives



1. **searching** for the best subset of features
2. **criteria** for evaluating different subsets
3. **principle for selecting**, adding, removing or changing new features during the search

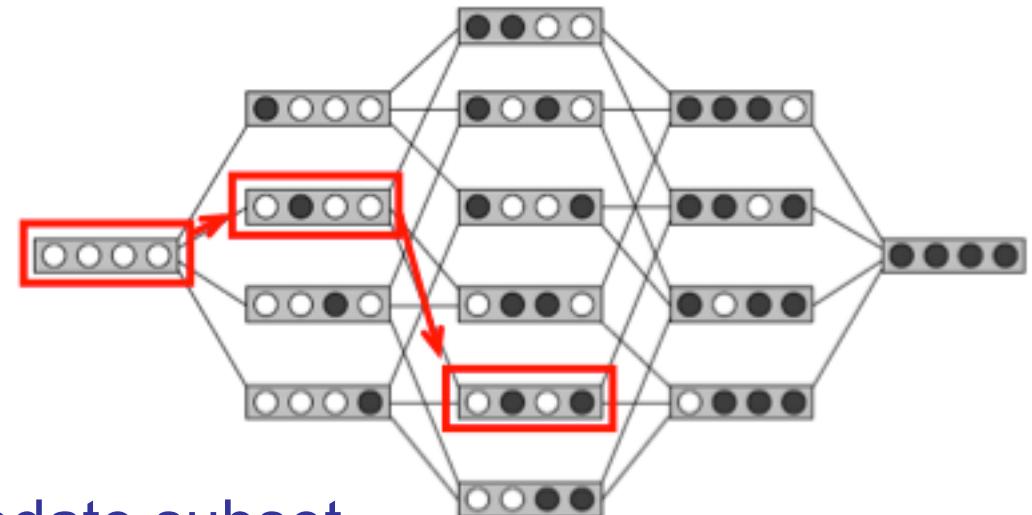


## Search of a Subset of Features

- **Search Directions:**
  - Sequential Forward Generation (SFG)
  - Sequential Backward Generation (SBG)
  - Bidirectional Generation (BG)
  - Random Generation (RG)
- **Search Strategies:**
  - Exhaustive Search
  - Heuristic Search
  - Non deterministic Search

# Sequential Forward Generation

- **Sequential Forward Generation (SFG):** It starts with an empty set of features S. As the search starts, features are added into S according to some criterion that distinguish the best feature from the others. S grows until it reaches a full set of original features. The stopping criteria can be a threshold for the number of relevant features m or simply the generation of all possible subsets in brute force mode.



- Define an initial subset
  - Begin with empty set
- Choose a strategy to update subset
- Decide when to stop

- Search Directions:

---

## Algorithm 1 Sequential forward feature set generation - SFG.

---

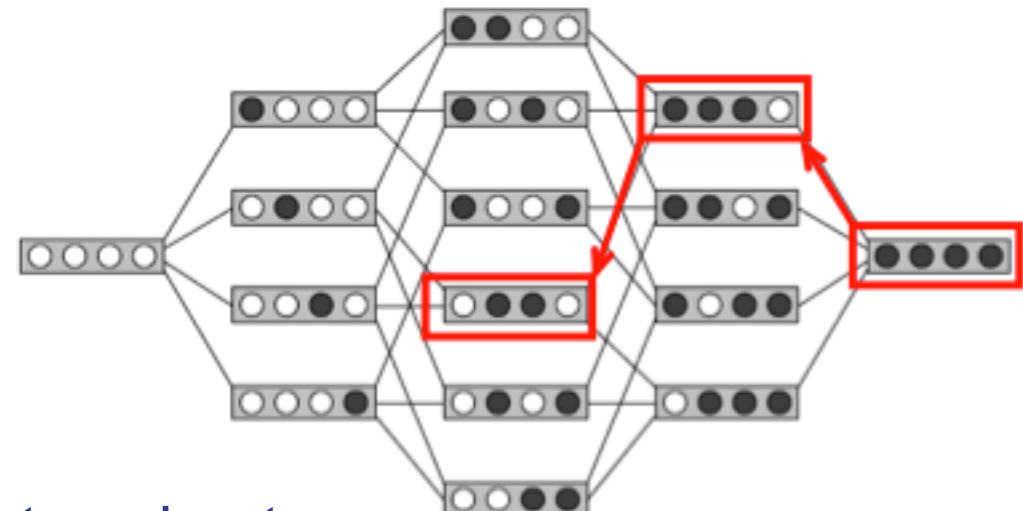
```
function SFG( $F$  - full set,  $U$  - measure)
    initialize:  $S = \{\}$                                  $\triangleright S$  stores the selected features
    repeat
         $f = \text{FINDNEXT}(F)$ 
         $S = S \cup \{f\}$ 
         $F = F - \{f\}$ 
    until  $S$  satisfies  $U$  or  $F = \{\}$ 
    return  $S$ 
end function
```

---

# Sequential Backward Generation

- **Sequential Backward Generation (SBG):** It starts with a full set of features and iteratively, they are removed one at a time. Here, the criterion must point out the worst or least important feature. By the end, the subset is only composed of a unique feature, which is considered to be the most informative of the whole set. As in the previous case, different stopping criteria can be used.

- Define an initial subset
  - Begin with full set



- Choose a strategy to update subset
- Decide when to stop

- Search Directions:

---

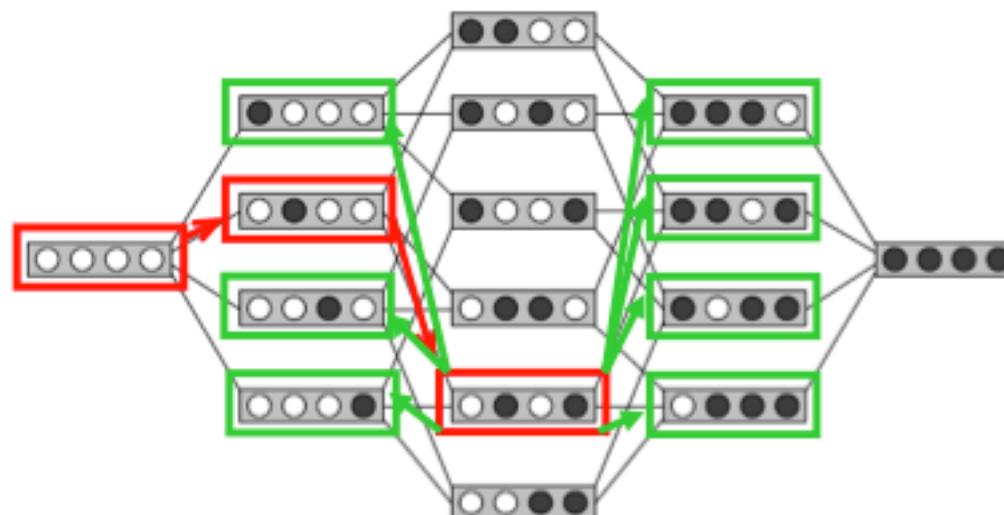
## Algorithm 2 Sequential backward feature set generation - SBG.

---

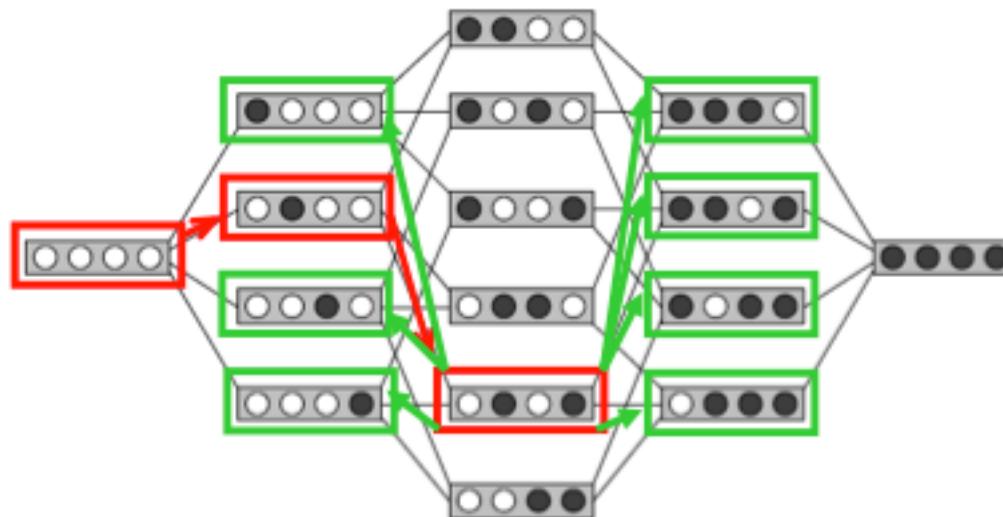
```
function SBG( $F$  - full set,  $U$  - measure)
    initialize:  $S = \{\}$                                  $\triangleright S$  holds the removed features
    repeat
         $f = \text{GETNEXT}(F)$ 
         $F = F - \{f\}$ 
         $S = S \cup \{f\}$ 
    until  $S$  does not satisfy  $U$  or  $F = \{\}$ 
    return  $F \cup \{f\}$ 
end function
```

---

- Forward-backward
  - Begins the search in both directions, performing SFG and SBG concurrently. They stop in two cases:
    - (1) when one search finds the best subset comprised of  $m$  features before it reaches the exact middle, or
    - (2) both searches achieve the middle of the search space. It takes advantage of both SFG and SBG.



- Forward-backward
  - At each step, consider all additions and removals of one variable, and select the best result



- Search Directions:

---

### Algorithm 3 Bidirectional feature set generation - BG.

---

**function**  $\text{BG}(F_f, F_b$  - full set,  $U$  - measure)

**initialize:**  $S_f = \{\}$

▷  $S_f$  holds the selected features

**initialize:**  $S_b = \{\}$

▷  $S_b$  holds the removed features

**repeat**

$f_f = \text{FINDNEXT}(F_f)$

$f_b = \text{GETNEXT}(F_b)$

$S_f = S_f \cup \{f_f\}$

$F_b = F_b - \{f_b\}$

$F_f = F_f - \{f_f\}$

$S_b = S_b \cup \{f_b\}$

**until** (a)  $S_f$  satisfies  $U$  or  $F_f = \{\}$  or (b)  $S_b$  does not satisfy  $U$  or  $F_b = \{\}$

**return**  $S_f$  if (a) or  $F_b \cup \{f_b\}$  if (b)

**end function**

- Search Directions:

---

## Algorithm 4 Random feature set generation - RG.

```
function RG( $F$  - full set,  $U$  - measure)
    initialize:  $S = S_{best} = \{\}$                                  $\triangleright S$  - subset set
    initialize:  $C_{best} = \#(F)$                                  $\triangleright \#$  - cardinality of a set
    repeat
         $S = \text{RANDGEN}(F)$ 
         $C = \#(S)$ 
        if  $C \leq C_{best}$  and  $S$  satisfies  $U$  then
             $S_{best} = S$ 
             $C_{best} = C$ 
        end if
    until some stopping criterion is satisfied
    return  $S_{best}$                                                $\triangleright$  Best set found so far
end function
```

- Search Strategies:

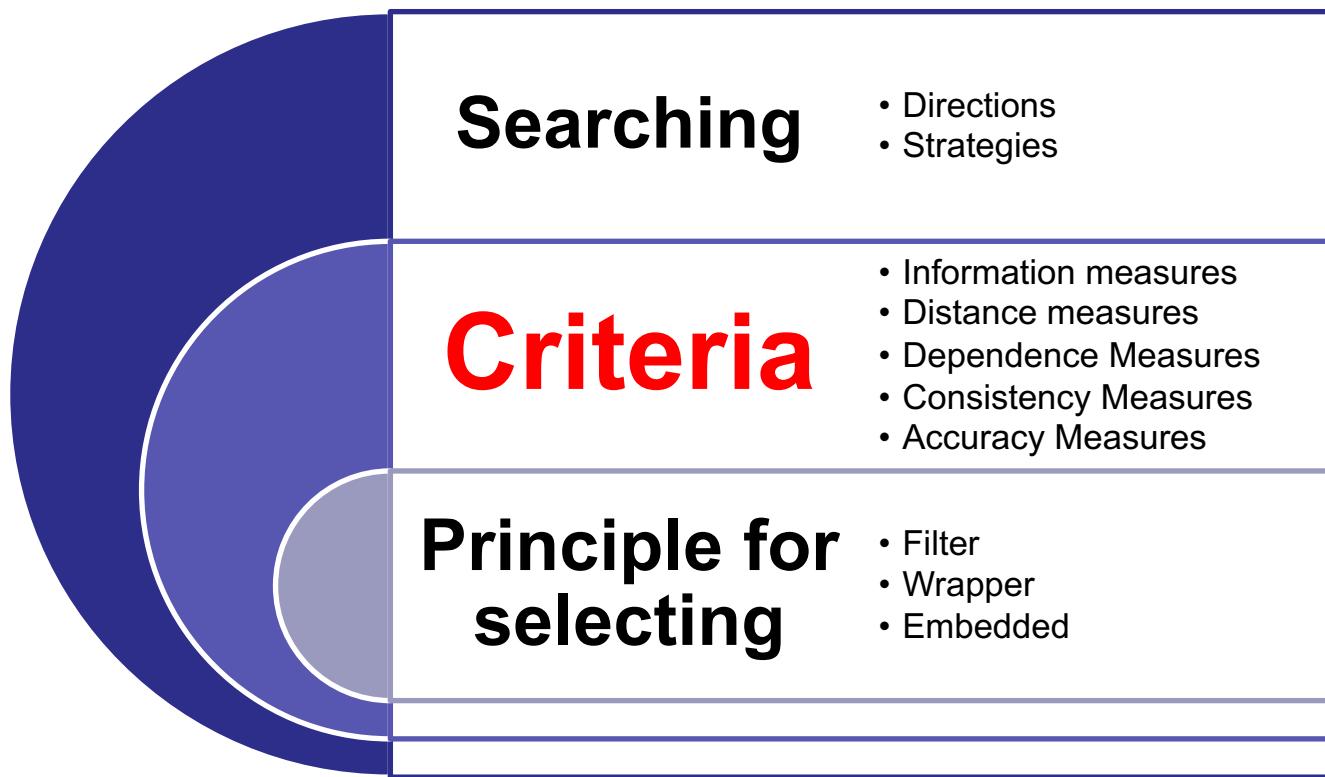
- **Exhaustive Search:** It corresponds to explore all possible subsets to find the optimal ones. As we said before, the space complexity is  $O(2^M)$ . If we establish a threshold  $m$  of minimum features to be selected and the direction of search, the search space is, independent of the forward or backward generation. Only exhaustive search can guarantee the optimality. Nevertheless, they are also impractical in real data sets with a high  $M$ .
- **Heuristic Search:** It employs heuristics to carry out the search. Thus, it prevents brute force search, but it will surely find a non-optimal subset of features. It draws a path connecting the beginning and the end of the previous Figure, such in a way of a depth-first search. The maximum length of this path is  $M$  and the number of subsets generated is  $O(M)$ . The choice of the heuristic is crucial to find a closer optimal subset of features in a faster operation.

- **Search Strategies:**

- **Non-deterministic Search:** Complementary combination of the previous two. It is also known as random search strategy and can generate best subsets constantly and keep improving the quality of selected features as time goes by. In each step, the next subset is obtained at random.

- it is unnecessary to wait until the search ends.
    - we do not know when the optimal set is obtained, although we know which one is better than the previous one and which one is the best at the moment.

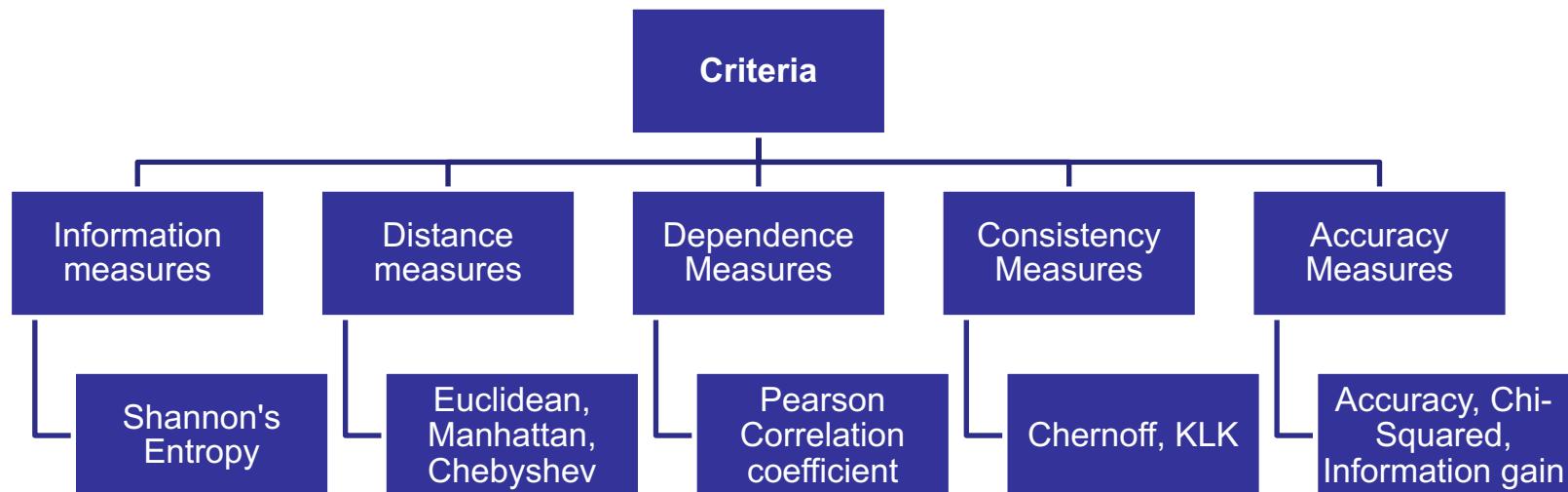
# Feature selection perspectives



1. **searching** for the best subset of features
2. **criteria** for evaluating different subsets
3. **principle for selecting**, adding, removing or changing new features during the search

- **Selection Criteria:**

- **Information Measures**
- **Distance Measures**
- **Dependence Measures**
- **Consistency Measures**
- **Accuracy Measures**



## – Information Measures.

- Information serves to measure the uncertainty of the receiver when she/he receives a message.
- **Shannon's Entropy:**

$$-\sum_i P(c_i) \log_2 P(c_i).$$

- **Information gain:**

$$IG(A) = I(D) - \sum_{j=1}^p \frac{|D_j|}{|D|} I(D_j^A)$$

## –Distance Measures.

- Measures of separability, discrimination or divergence measures. The most typical is derived from distance between the class conditional density functions.

	Mathematical form
Euclidean distance	$D_e = \left\{ \sum_{i=1}^m (x_i - y_i)^2 \right\}^{\frac{1}{2}}$
City-block distance	$D_{cb} = \sum_{i=1}^m  x_i - y_i $
Cebyshев distance	$D_{ch} = \max_i  x_i - y_i $
Minkowski distance of order $m$	$D_M = \left\{ \sum_{i=1}^m (x_i - y_i)^m \right\}^{\frac{1}{m}}$
Quadratic distance $Q$ , positive definite	$D_q = \sum_{i=1}^m \sum_{j=1}^m (x_i - y_i) Q_{ij} (x_j - y_j)$
Canberra distance	$D_{ca} = \sum_{i=1}^m \frac{ x_i - y_i }{x_i + y_i}$
Angular separation	$D_{as} = \frac{\sum_{i=1}^m x_i \cdot y_i}{[\sum_{i=1}^m x_i^2 \sum_{i=1}^m y_i^2]^{\frac{1}{2}}}$

## – Dependence Measures.

- known as measures of association or correlation.
- Its main goal is to quantify how strongly two variables are correlated or present some association with each other, in such way that knowing the value of one of them, we can derive the value for the other.
- **Pearson correlation coefficient:**

$$\rho(X, Y) = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\left[ \sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2 \right]^{\frac{1}{2}}}$$

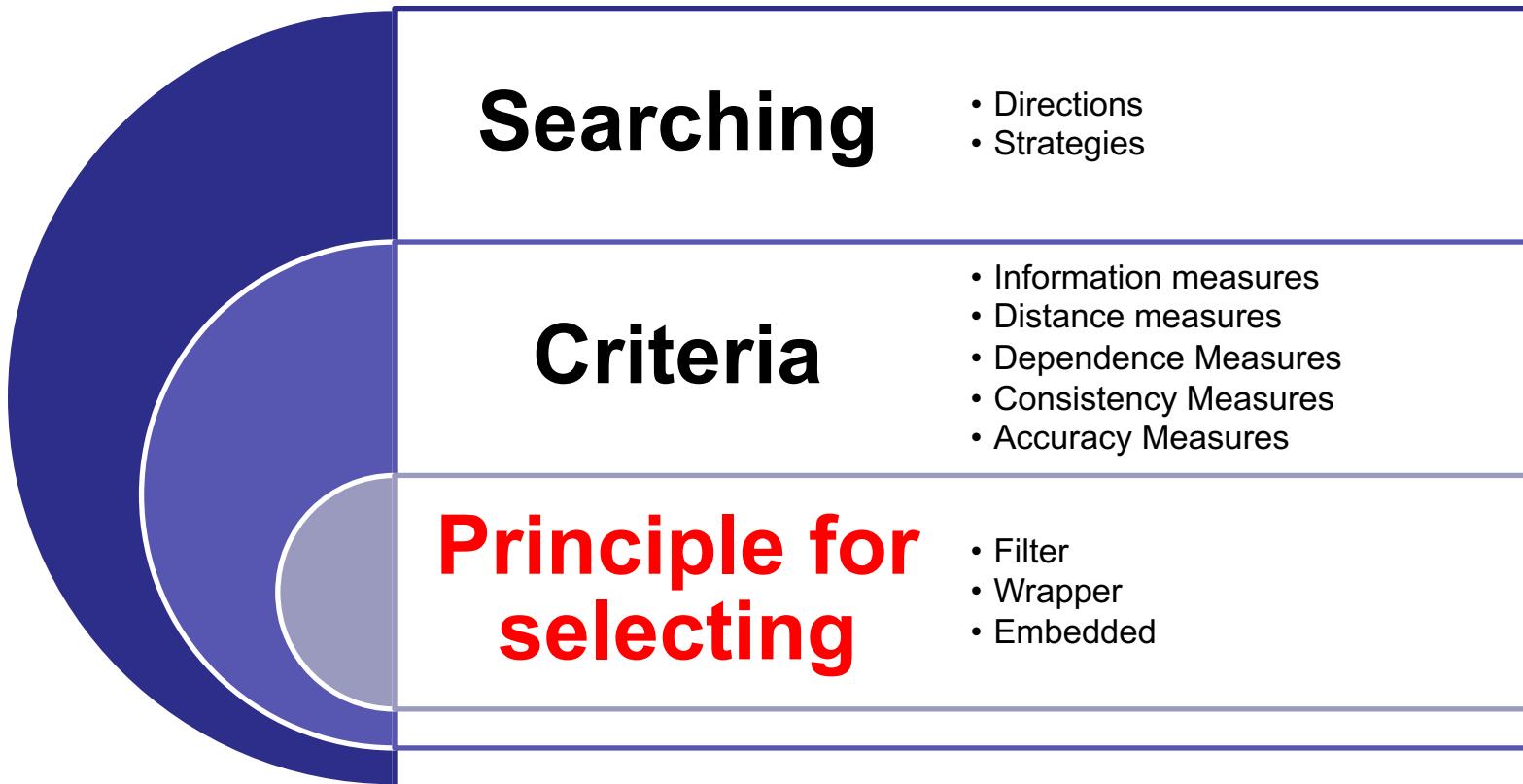
## –Consistency Measures.

- They attempt to find a minimum number of features that separate classes as the full set of features can.
- They aim to achieve  $P(C|FullSet) = P(C|SubSet)$ .
- An inconsistency is defined as the case of two examples with the same inputs (same feature values) but with different output feature values (classes in classification).

## – Accuracy Measures.

- This form of evaluation relies on the classifier or learner. Among various possible subsets of features, the subset which yields the best predictive accuracy is chosen

	Mathematical form
Accuracy	$\frac{tp + tn}{tp + tn + fp + fn}$
Error rate	1 – Accuracy
Chi-squared	$\frac{n(fp \times fn - tp \times tn)^2}{(tp+fp)(tp+fn)(fp+tn)(tn+fn)}$
Information gain	$e(tp + fn, fp + tn) - \frac{(tp+fp)e(tp, fp) + (tn+fn)e(fn, tn)}{tp+fp+tn+fn}$ where $e(x, y) = -\frac{x}{x+y} \log_2 \frac{x}{x+y} - \frac{y}{x+y} \log_2 \frac{y}{x+y}$
Odds ratio	$\frac{tpr}{1-tpr} / \frac{fpr}{1-fpr} = \frac{tp \times tn}{fp \times fn}$
Probability ratio	$\frac{tpr}{fpr}$

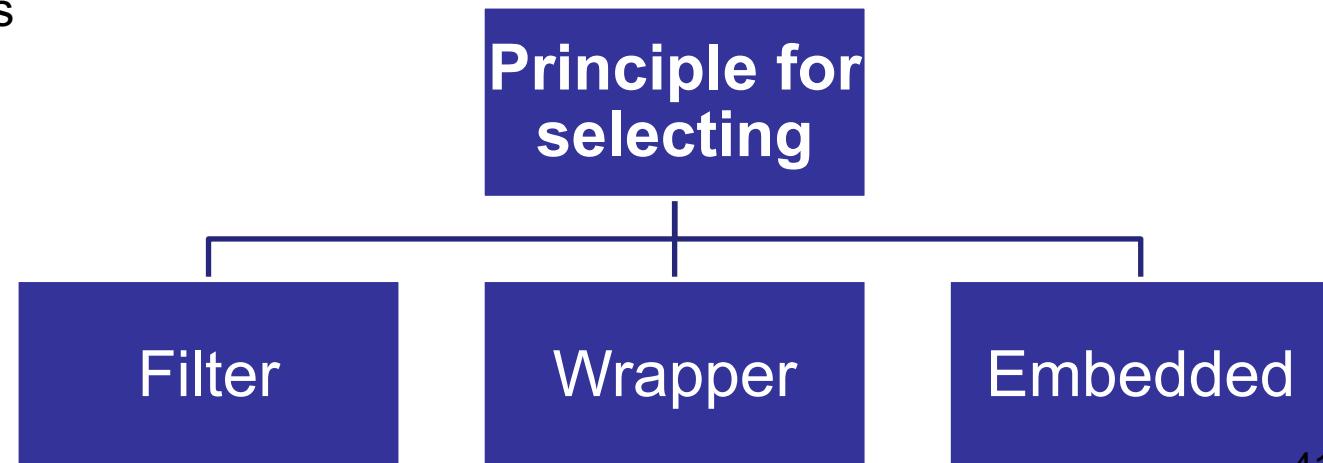


1. **searching** for the best subset of features
2. **criteria** for evaluating different subsets
3. **principle for selecting**, adding, removing or changing new features during the search

# Feature selection perspectives

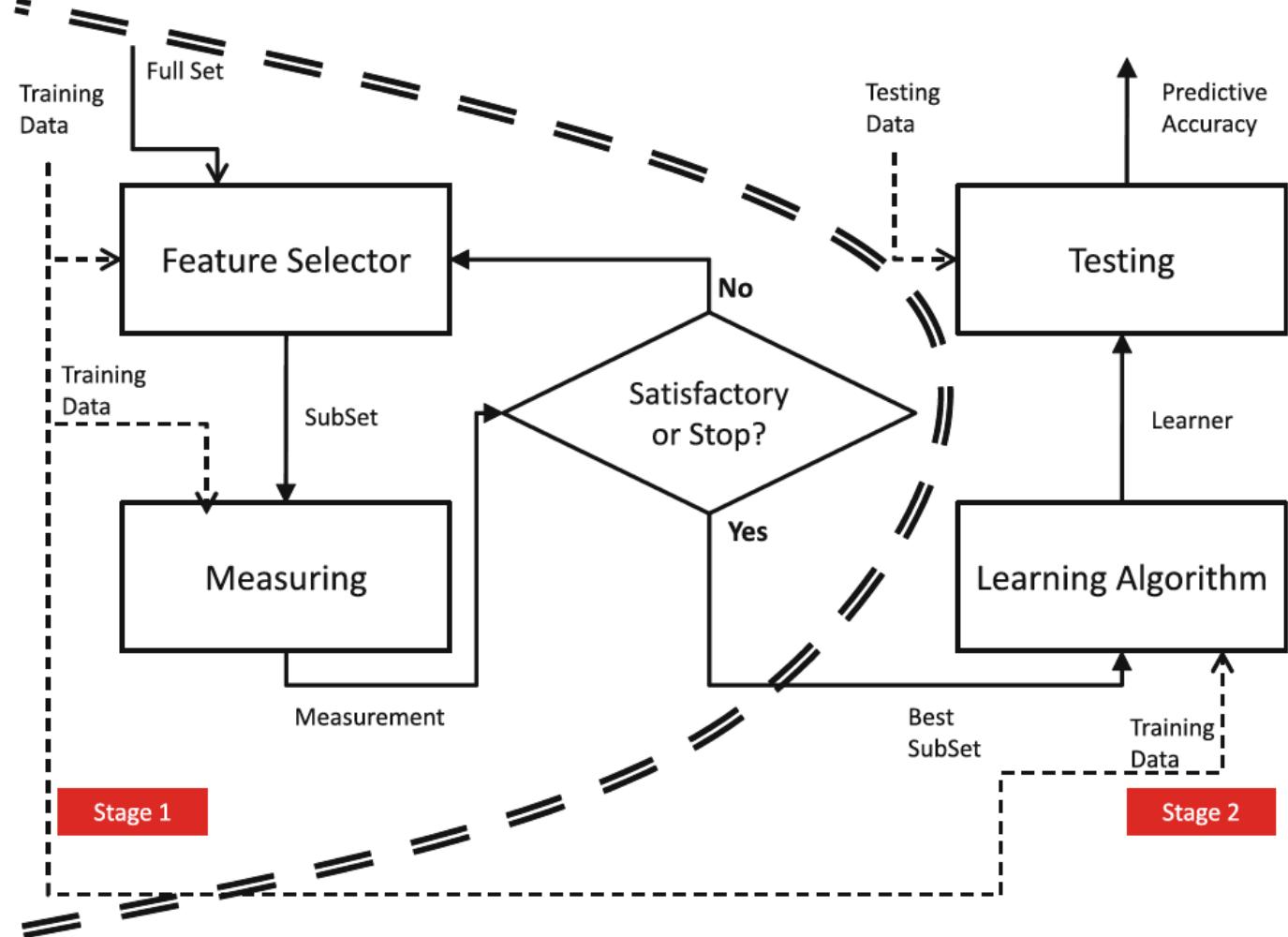
Generally are classified according to the criterion function used in searching for good features

1. **Filter algorithm**: some feature evaluation function is used rather than optimizing the classifier's performance.
2. **Wrapper algorithm**: the performance of the classifier is used to evaluate the feature subsets.
3. **Embedded feature selection algorithm**: performs variable selection (implicitly) in the course of model training. Similar to wrappers, but in this approach, the features are selected during the learning process



# Feature selection perspectives

- **Filters:**

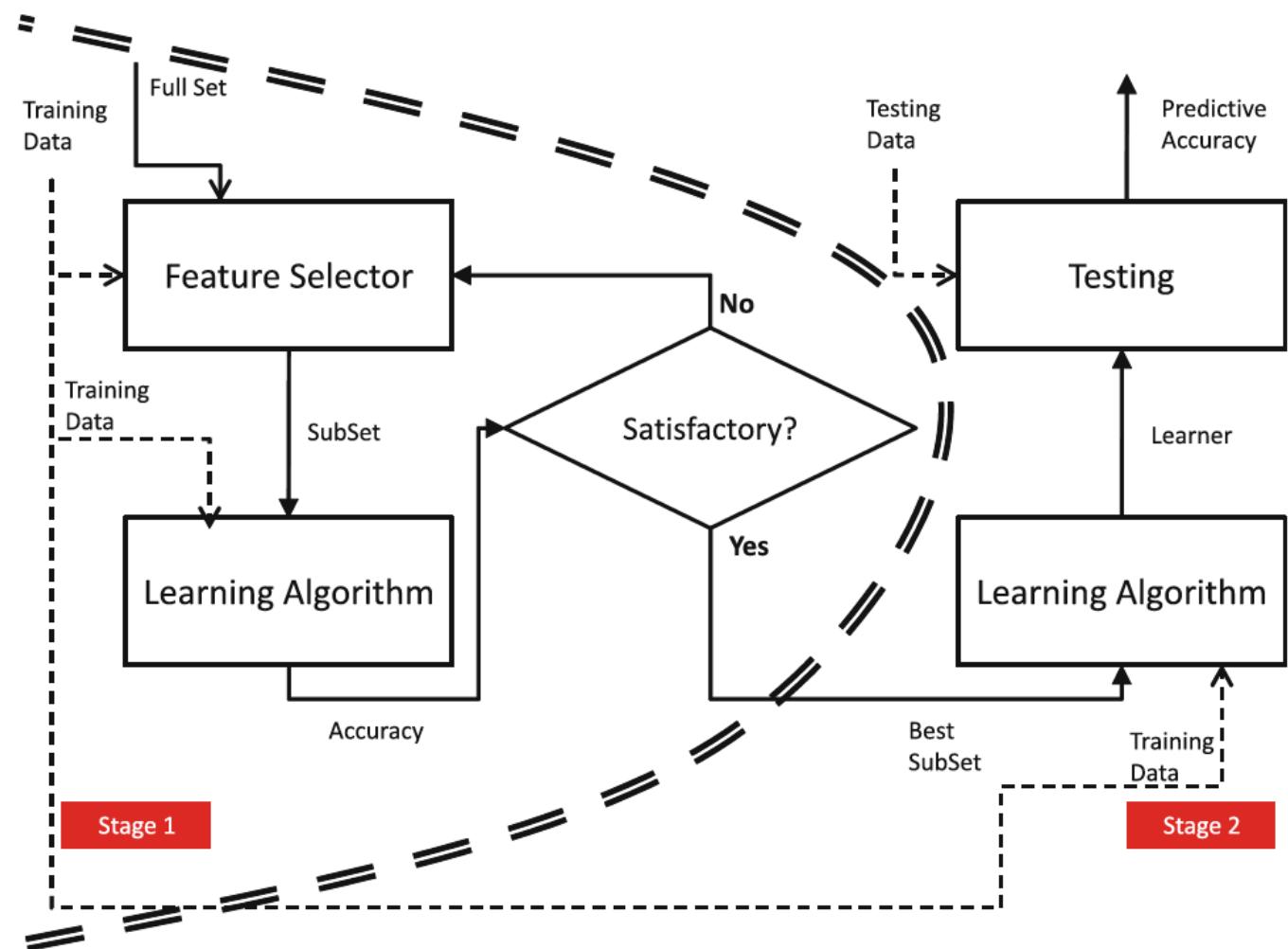


- **Filters:**

- measuring uncertainty, distances, dependence or consistency is usually cheaper than measuring the accuracy of a learning process. Thus, filter methods are usually faster.
- it does not rely on a particular learning bias, in such a way that the selected features can be used to learn different models from different DM techniques.
- it can handle larger sized data, due to the simplicity and low time complexity of the evaluation measures.

# Feature selection perspectives

- **Wrappers:**



- **Wrappers:**

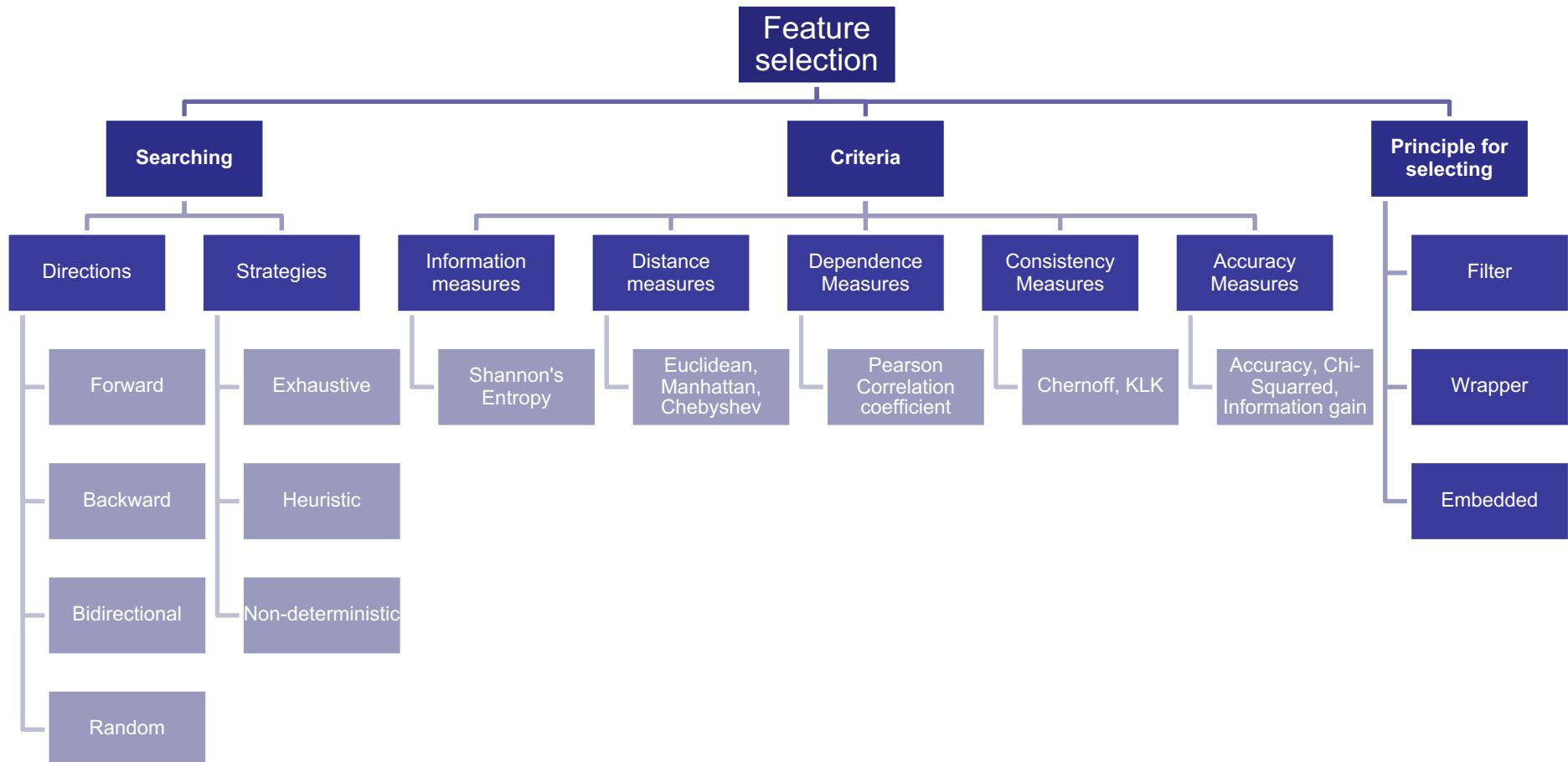
- can achieve the purpose of improving the particular learner's predictive performance.
- usage of internal statistical validation to control the overfitting, ensembles of learners and hybridizations with heuristic learning like Bayesian classifiers or Decision Tree induction.
- filter models cannot allow a learning algorithm to fully exploit its bias, whereas wrapper methods do.

- **Embedded FS:**
  - similar to the wrapper approach in the sense that the features are specifically selected for a certain learning algorithm, but in this approach, the features are selected during the learning process.
  - they could take advantage of the available data by not requiring to split the training data into a training and validation set; they could achieve a faster solution by avoiding the re-training of a predictor for each feature subset explored.

# Wrappers vs. Filters

FILTERS	WRAPPERS
<ul style="list-style-type: none"><li>+ <b>Fast</b>: build only one model</li><li>+ <b>Intuitive</b>: identifies statistical dependency</li></ul>	<ul style="list-style-type: none"><li>+ Relevance criterion <b>easy</b> to estimate</li><li>+ <b>Model-aware</b>: identifies optimal subset to build optimal model</li></ul>
<ul style="list-style-type: none"><li>- Relevance criterion <b>hard to estimate</b></li><li>- <b>Model-ignorant</b>: most relevant subset might not be optimal for subsequent model</li></ul>	<ul style="list-style-type: none"><li>- <b>Slow</b>: must build lots of models</li><li>- <b>Not intuitive</b>: features for best model might not actually be most explanatory variables</li></ul>

# Summary feature selection perspectives



# Aspects in feature selection

- ## Feature Ranking Techniques:

- we expect as the **output a ranked list of features** which are ordered according to evaluation measures.
- they return the **relevance of the features**.
- For performing actual FS, the simplest way is to choose the first  $m$  features for the task at hand, whenever we know the most appropriate  $m$  value.

- **Feature Ranking Techniques:**

---

**Algorithm 5** A univariate feature ranking algorithm.

---

```
function RANKING ALGORITHM( $x$  - features,  $U$  - measure)
    initialize: list  $L = \{\}$                                      ▷  $L$  stores ordered features
    for each feature  $x_i, i \in \{1, \dots, M\}$  do
         $v_i = \text{COMPUTE}(x_i, U)$ 
        position  $x_i$  into  $L$  according to  $v_i$ 
    end for
    return  $L$  in decreasing order of feature relevance.
end function
```

---

- **Minimum Subset Techniques:**

- The number of relevant features is a parameter that is often not known by the practitioner.
- There must be a second category of techniques focused on obtaining the minimum possible subset without ordering the features.
- whatever is relevant within the subset, is otherwise irrelevant.

- **Minimum Subset Techniques:**

---

## Algorithm 6 A minimum subset algorithm.

---

```
function MIN-SET ALGORITHM(x - features, U - measure)
    initialize: L = {}, stop = false                                ▷ S holds the minimum set
    repeat
        Sk = SUBSETGENERATE(x)
        if LEGITIMACY(Sk, U) is true and #(Sk) < #(S) then
            S = Sk                                              ▷ stop can be set here
        end if
    until stop = true
    return S - the minimum subset of features
end function
```

---

- **Goals:**
  - **Inferability:** For predictive tasks, considered as an improvement of the prediction of unseen examples with respect to the direct usage of the raw training data.
  - **Interpretability:** Given the incomprehension of raw data by humans, DM is also used for generating more understandable structure representation that can explain the behavior of the data.
  - **Data Reduction:** It is better and simpler to handle data with lower dimensions in terms of efficiency and interpretability.

- We can derive several assessment measures from these three goals:
  - Accuracy
  - Complexity
  - Number of Features Selected
  - Speed of the FS method
  - Generality of the features selected

# Aspects: Drawbacks

- The resulted subsets of many models of FS are strongly dependent on the training set size
- It is not true that a large dimensionality input can always be reduced to a small subset of features because the objective feature is actually related with many input features and the removal of any of them will seriously affect the learning performance
- A backward removal strategy is very slow when working with large-scale data sets. This is because in the first stages of the algorithm, it has to make decisions based on huge quantities of data
- In some cases, the FS outcome will still be left with a relatively large number of relevant features which even inhibit the use of complex learning methods



# Most representative feature selection methods

# Most Representative Methods

- Three major components to categorize combinations:
  - **Search Direction**
  - **Search Strategy**
  - **Evaluation Measure**

Search direction	Evaluation measure	Search strategy		
		Exhaustive	Heuristic	Nondeterministic
Forward	Probability	C1	C7	—
	Consistency	C2	C8	—
	Accuracy	C3	C9	—
Backward	Probability	C4	C10	—
	Consistency	C5	C11	—
	Accuracy	C6	C12	—
Random	Probability	—	C13	C16
	Consistency	—	C14	C17
	Accuracy	—	C15	C18

## Exhaustive Methods

- Cover the whole search space.
- Six Combinations (C1-C6).
  - **Focus method**: C2.
  - Automatic Branch and Bound (ABB): C5.
  - Best First Search (BFS): C1.
  - Beam Search: C3.
  - Branch and Bound (BB): C4.

---

## Algorithm 7 Focus algorithm.

---

```
function FOCUS( $F$  - all features in data  $D$ ,  $U$  - inconsistency rate as evaluation measure)
    initialize:  $S = \{\}$ 
    for  $i = 1$  to  $M$  do
        for each subset  $S$  of size  $i$  do
            if  $\text{CALU}(S,D) = 0$  then
                return  $S$  - a minimum subset that satisfies  $U$ 
            end if
        end for
    end for
end function
```

▷  $\text{CalU}(S,D)$  returns inconsistency

## Heuristic Methods

- They do not have any expectations of finding an optimal subset with a rapid solution.
- Nine Combinations (C7-C15).
  - Use a DM algorithm for FS: C12.
  - Wrapper Sequential Forward Selection: C9.
  - SetCover: C8.
  - Heuristic search algorithm and in each sub-search space: C13-C15.
  - **MIFS: C10.**

---

## Algorithm 8 MIFS algorithm.

```
function MIFS( $F$  - all features in data,  $S$  - set of selected features,  $k$  - desired size of  $S$ ,  $\beta$  - regulator parameter)
    initialize:  $S = \{\}$ 
    for each feature  $f_i$  in  $F$  do
        Compute  $I(C, f_i)$ 
    end for
    Find  $f_{max}$  that maximizes  $I(C, f)$ 
     $F = F - \{f_{max}\}$ 
     $S = S \cup f_{max}$ 
    repeat
        for all couples of features ( $f_i \in F, s_j \in S$ ) do
            Compute  $I(f_i, s_j)$ 
        end for
        Find  $f_{max}$  that maximizes  $I(C, f) - \beta \sum_{s \in S} I(f_i, s_j)$ 
         $F = F - \{f_{max}\}$ 
         $S = S \cup f_{max}$ 
    until  $|S| = k$ 
    return  $S$ 
```

## Nondeterministic Methods

- They add or remove features to and from a subset without a sequential order.
- Three Combinations (C16-C18).
  - Simulated Annealing / Genetic Algorithms are the most common techniques.
  - **LVF: C17.**
  - **LVW: C18.**

# Most Representative Methods

---

## Algorithm 9 LVF algorithm.

```

function LVF( $D$  - a data set with  $M$  features,  $U$  - the inconsistency rate,  $maxTries$  - stopping
criterion,  $\gamma$  - an allowed inconsistency rate)
initialize: list  $L = \{\}$  ▷  $L$  stores equally good sets
 $C_{best} = M$ 
for  $maxTries$  iterations do
     $S = \text{RANDOMSET}(\text{seed})$ 
     $C = \#(S)$  ▷  $\#$  - the cardinality of  $S$ 
    if  $C < C_{best}$  and  $\text{CALU}(S,D) < \gamma$  then
         $S_{best} = S$ 
         $C_{best} = C$ 
         $L = \{S\}$  ▷  $L$  is reinitialized
    else if  $C = C_{best}$  and  $\text{CALU}(S,D) < \gamma$  then
         $L = \text{APPEND}(S,L)$ 
    end if
end for
return  $L$  ▷ all equivalently good subsets found by LVF
end function

```

## Algorithm 10 LVW algorithm.

```
function LVW( $D$  - a data set with  $M$  features,  $LA$  - a learning algorithm,  $maxTries$  - stopping criterion,  $F$  - a full set of features)
    initialize: list  $L = \{\}$                                       $\triangleright L$  stores sets with equal accuracy
     $A_{best} = \text{ESTIMATE}(D, F, LA)$ 
    for  $maxTries$  iterations do
         $S = \text{RANDOMSET}(\text{seed})$ 
         $A = \text{ESTIMATE}(D, S, LA)$                                  $\triangleright \#$  - the cardinality of  $S$ 
        if  $A > A_{best}$  then
             $S_{best} = S$ 
             $A_{best} = A$ 
             $L = \{S\}$                                           $\triangleright L$  is reinitialized
        else if  $A = A_{best}$  then
             $L = \text{APPEND}(S, L)$ 
        end if
    end for
    return  $L$                                           $\triangleright$  all equivalently good subsets found by LVW
end function
```

## Feature Weighting Methods

- Provide weights to features, also can be used for FS.
- Relief (binary) and ReliefF (multiple classes).

---

### Algorithm 11 Relief algorithm.

```
function RELIEF(x - features, m - number of instances sampled, τ - relevance threshold)
    initialize: w = 0
    for i = 1 to m do
        randomly select an instance I
        find nearest-hit H and nearest-miss J
        for j = 1 to M do
            w(j) = w(j) - dist(j, I, H)2/m + dist(j, I, J)2/m    ▷ dist is a distance function
        end for
    end for
    return w greater than τ
end function
```



# Summary

- Feature selection has different elements
  - Searching for the best subset of features
    - Forward, Backward, Bidirectional, or Random generation
  - Selection Criteria
    - Heuristic, Exhaustive, or Non deterministic search
  - Criteria for evaluating different subsets
    - Information, distance, dependency, or accuracy measures
  - Principle for selecting (selection algorithms)
    - Filters, Wrappers, or Embedded algorithms



# What else?



## TO READ:

Guyon, I., Elisseeff, A., 2003. ***An Introduction to Variable and Feature Selection.***



*Journal Machine Learning Research 3, 1157–1182.*

- *Section 1, 2, 3, 4.1 are mandatory*
- *The remaining of the article is optional*

## Week 8

# Course. Introduction to Machine Learning

## Theory 8. Feature Selection

Dr. Maria Salamó Llorente

[maria.salamo@ub.edu](mailto:maria.salamo@ub.edu)

Dept. Mathematics and Informatics,  
Faculty of Mathematics and Informatics,  
University of Barcelona (UB)