Feature Subset Selection

FSS

Pedro Larrañaga, Concha Bielza

Computational Intelligence Group
Artificial Intelligence Department
Universidad Politécnica de Madrid

Main issue: detect the features that are relevant for the task but non-redundant (relevancy and non-redundancy)

Computational Intelligence Group

Departamento Inteligencia Artificial

Machine Learning

Master in Data Science + Master in HMDA

Outline

Outline

Feature Subset Selection

Feature subset selection (FSS) (Sebestyen, 1962; Lewis, 1962): identify and remove as many irrelevant and redundant variables as possible

People say that "less is more" -> less variables and less complex models

Advantages and disadvantages

- Reduction of the dimensionality of the data
- Helping the learning algorithms to operate faster and more effectively
- Improving the accuracy of the classifier Specially with the wrapper approach
- Improving the interpretation of the learned model If there are less variables, the model will be easier to interpret
- The price to be paid: computational burden

Feature Subset Selection

Relevant and redundant

- A discrete feature X_i is said to be a relevant feature for the class variable C iff there exists some x_i and c for which $p(X_i = x_i) > 0$ such that $p(C = c|X_i = x_i) \neq \underbrace{p(C = c)}_{\text{A priori distribution of C}}$
- A feature is said to be a redundant feature if it is highly correlated with one or more of the other features

Relevant and redundant for k-NN and naive Bayes

- The k nearest neighbour algorithm is sensitive to irrelevant variables
- The naive Bayes classifier can be negatively affected by redundant variables

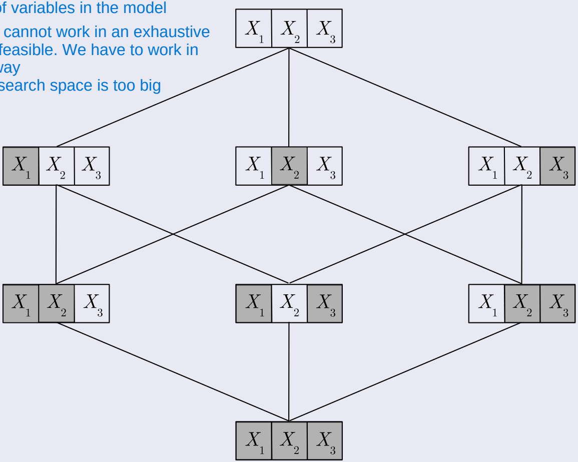
Feature Subset Selection

FSS can be seen as an optimization problem

Cardinality of the search space: 2^n

n = number of variables in the model

If n is big, we cannot work in an exhaustive way, it is not feasible. We have to work in an heuristic way because the search space is too big



Search space for an FSS problem with three predictor variables. Each of the eight blocks represent one possible FSS. The filled rectangles in each block indicate the variables included in the selected subset

Feature Subset Selection

The FSS problem consists of selecting the optimal subset $\mathcal{S}^* \subseteq \mathcal{X} = \{X_1, ..., X_n\}$ with respect to an objective score that, without loss of generality, should be maximized

Notation. Objective score

$$\begin{array}{rccc} f: & \mathcal{P}(\mathcal{X}) & \longrightarrow & \mathbb{R} \\ & \mathcal{S} \subseteq \mathcal{X} & \longmapsto & f(\mathcal{S}), \end{array}$$

$\mathcal{P}(\mathcal{X})$ denotes the set of all possible subsets of $\mathcal{X}$, whose cardinality is given by 2^n

Notation. Representing FSS solutions

Binary vector $\mathbf{s} = (s_1, ..., s_n)$, with

$$s_i = \begin{cases} 1 & \text{if variable } X_i \text{ belongs to } \mathcal{S} \\ 0 & \text{otherwise} \end{cases}$$

Notation. The optimal FSS

$$\begin{array}{rccc} f: & \{0,1\}^n & \longrightarrow & \mathbb{R} \\ & \mathbf{s} = (s_1, ..., s_n) & \longmapsto & f(\mathbf{s}). \end{array}$$

The optimal feature subset, $\mathbf{s}^*$, verifies $s^* = \arg\max_{\mathbf{s} \in \{0,1\}^n} f(\mathbf{s})$

Feature Subset Selection

Characteristics affecting the nature of the search

(a) ***Starting point***

- No features For example, we can add one variable in each step
- All features Now, we delete one variable at a time
- A subset of features

(b) ***Search organisation***

We decide the methodology according to what we think: if we think that the subset is going to be small, use forward. On the other hand, if we think that we are going to keep almost every feature, use backward

- Exhaustive If n is not very large, we can try all the possibilities
- Forward We start from scratch and in each step we introduce one variable until some stopping criteria. In backward, we delete one variable
- Backward
- Stepwise More flexible that forward and backward
- Based on metaheuristics

(c) ***Evaluation strategy***

- Filter Only looking for intrinsic characteristics of the data, not considering a model
- Wrapper Need to evaluate each subset with the metric we are considering (accuracy, F1-score...) -> very time consuming

(d) ***Stopping criterion***

- Until no improvement of the objective function

Outline

Filter feature subset selection

Filter feature subset selection methods assess the relevance of a feature (univariate filtering), or a subset of features (multivariate filtering), by looking only at intrinsic properties of the data

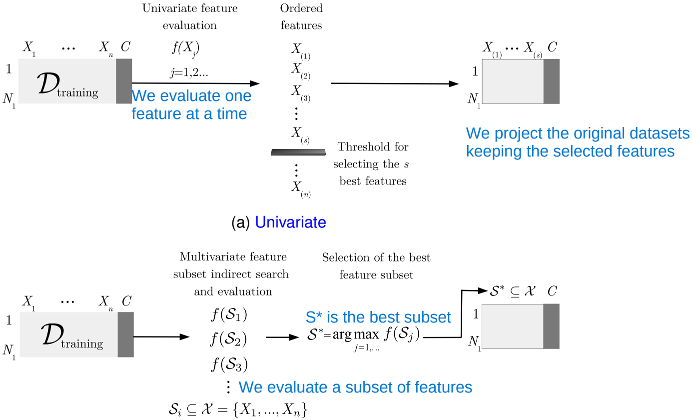
We don't need to build any classifier to select the subset

Advantages

- They easily scale to very high-dimensional data sets
- They are computationally simple and fast
- They avoid overfitting problems
- They are independent of the supervised classification algorithm

  The result can be used for any classifier
- Filter feature selection needs to be performed only once. This selection is evaluated later with different classification models

Univariate versus multivariate filtering



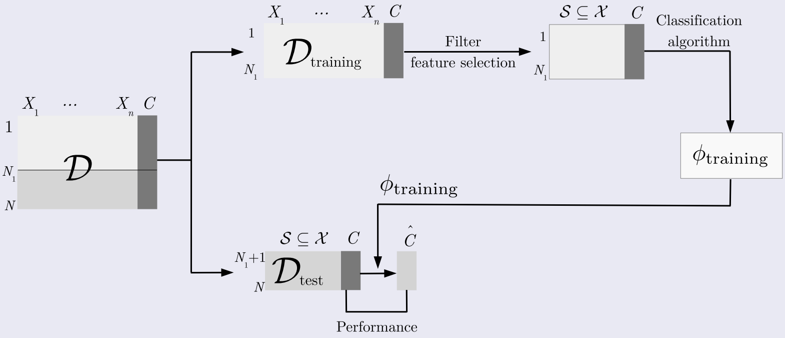
(a) Univariate

(b) Multivariate

(a) Univariate filter: original variables $X_1, \ldots, X_n$ are ordered according to $f(X_1), \ldots, f(X_n)$ resulting in the ordered variables $X_{(1)}, \ldots, X_{(n)}$. A threshold chooses the s best variables of that ranking, which is the final feature subset on which to start the classifier learning. (b) Multivariate filter: a subset of features $\mathcal{S}$ is searched and evaluated according to $f(\mathcal{S})$. The best subset $\mathcal{S}^*$ is found as an optimization problem and this is the final feature subset on which to start the classifier learning

Evaluation of a Classification Model Output by Filter FSS

Hold-out scheme



Univariate Filtering Methods

Parametric methods: We assume a probability distribution over the variables	
Discrete predictors:	
Mutual information	Blanco et al. (2005)
Gain ratio	Hall and Smith (1998)
Symmetrical uncertainty	Hall (1999)
Chi-squared	Forman (2003)
Odds ratio	Mladenic and Grobelnik (1999)
Bi-normal separation	Forman (2003
Continuous predictors:	
t-test family	Jafari and Azuaje (2006)
ANOVA	Jafari and Azuaje (2006)
Model-free methods:	
Threshold number of misclassification (TNoM)	Ben-dor et al. (2000)
P-metric	Slonim et al. (2000)
Mann-Whitney test	Thomas et al. (2001)
Kruskal-Wallis test	Lan and Vucetic (2011)
Between-groups to within-groups sum of squares	Dudoit et al. (2002)
Scores based on estimating density functions	Inza et al. (2004)

Univariate Filter. Parametric

Discrete predictors. Mutual information. Gain ratio. Symmetrical uncertainty

The mutual information between two variables X_j and C:

$$f(X_j) = \mathbb{I}(X_j, C) = -\sum_{i=1}^{R_j}\sum_{c=1}^{R} p(X_j = i, C = c)\log_2 \underbrace{p(X_j = i, C = c)}_{\text{p(X\_j = i) * p(C=c)}}$$

(errata)

- Under the null hypothesis of independence between X_j and C, the statistic $2N\mathbb{I}(X_j, C) \sim \chi^2_{(R_j-1)(R-1)}$
- Select the predictor variables with the k highest mutual information values, where k was fixed according to the p-values
- Variables with small p-values (where the null hypothesis of independence is rejected) are selected as relevant for the class variable
- The mutual information measure favors variables with many different values over others with few different values. A fairer selection is to use gain ratio defined as $\frac{\mathbb{I}(X_j, C)}{\mathbb{H}(X_j)}$ or the symmetrical uncertainty coefficient defined as $2\frac{\mathbb{I}(X_j, C)}{\mathbb{H}(X_j)+\mathbb{H}(C)}$

Univariate Filter. Parametric

Discrete predictors. Chi-squared

Chi-squared based feature selection measures the divergence from the distribution expected if one assumes that feature occurrence is actually independent of the class value

$$
\begin{array}{cc|cc|c}
 & & \multicolumn{2}{c}{C} & \\
 & & 1 & 2 & \text{Marginal} \\
\hline
X_j & 1 & N_{11} & N_{12} & N_{1\bullet} \\
 & 2 & N_{21} & N_{22} & N_{2\bullet} \\
\hline
 & \text{Marginal} & N_{\bullet 1} & N_{\bullet 2} & N
\end{array}
$$

$$f(X_j) = \frac{(N_{11} - \frac{N_{1\bullet}N_{\bullet 1}}{N})^2}{\frac{N_{1\bullet}N_{\bullet 1}}{N}} + \frac{(N_{12} - \frac{N_{1\bullet}N_{\bullet 2}}{N})^2}{\frac{N_{1\bullet}N_{\bullet 2}}{N}} + \frac{(N_{21} - \frac{N_{2\bullet}N_{\bullet 1}}{N})^2}{\frac{N_{2\bullet}N_{\bullet 1}}{N}} + \frac{(N_{22} - \frac{N_{2\bullet}N_{\bullet 2}}{N})^2}{\frac{N_{2\bullet}N_{\bullet 2}}{N}}$$

- Features are ranked in ascending order according to their p-value. The variables most dependent on the class (smallest p-values) rank first
- After fixing a threshold for the p-value, the classifier will only take into account variables with p-values smaller than the threshold

Univariate Filter. Model-free

Mann-Whitney test + Kruskal-Wallis test

- The Mann-Whitney test based method for testing the equality of two population means in two unpaired samples. Variables are sorted according to their *p*-values. Small *p*-values are ranked highest
- The Kruskal-Wallis test based method for testing the equality of more than two population means from unpaired samples

Multivariate Filter

Multivariate filtering methods

RELIEF	Kira and Rendell (1992)
Correlation-based feature selection	Hall (1999)
Conditional mutual information	Fleuret (2004)

Multivariate Filter

It has more than 50 variants. Not only used for supervised classification but for regression, too

Not easy to understand, try to familiarise with the pseudocode

RELIEF

Algorithm 1: The RELIEF algorithm

Input : A data set $\mathcal{D}$ of N labelled instances, a vector $\mathbf{w} = (w_1, ..., w_n)$ initialized as $(0, ..., 0)$

W_i corresponds to variable i

Output: The vector $\mathbf{w}$ of the relevancies estimates of the n predictor variables

for $i = 1$ **to** N **do**
    Randomly select an instance $\mathbf{x} \in \mathcal{D}$
    Find near-hit $\mathbf{x}^h \in \mathcal{D}$, and near-miss $\mathbf{x}^m \in \mathcal{D}$
    for $j = 1$ **to** n **do**
        $w_j = w_j - \frac{1}{N} d_j(\mathbf{x}, \mathbf{x}^h) + \frac{1}{N} d_j(\mathbf{x}, \mathbf{x}^m)$
    endfor
endfor

near-hit: closest example to x with the same label
near-miss: closes example with different label

Multivariate Filter

Correlation-based feature selection (CFS)

CFS seeks for a feature subset that contains features that are highly correlated with the class, yet uncorrelated with each other

- $\mathcal{S}^* = \arg\max_{\mathcal{S} \subseteq \mathcal{X}} f(\mathcal{S})$, where

$$f(\mathcal{S}) = \frac{\sum_{X_i \in \mathcal{S}} r(X_i, C)}{\sqrt{k + (k-1) \sum_{X_i, X_j \in \mathcal{S}} r(X_i, X_j)}}$$

  - k is the number of selected features,
  - $r(X_i, C)$ is the correlation between feature X_i and class variable C
  - $r(X_i, X_j)$ is the correlation between features X_i and X_j
  - $r(X_i, C)$ is given by the symmetrical uncertainty coefficient
- In the initial proposal three heuristic search strategies: forward selection, backward elimination, and best-first search
- Other metaheuristics like tabu search, variable neighbor search, genetic algorithms and estimation of distribution algorithms, among others, have been applied for CFS

Multivariate Filter

Conditional mutual information

- Feature ranking criterion based on conditional mutual information for binary data based on the idea that feature X_i is good only if $\mathbb{I}(X_i, C|X_j)$ is large for every already selected X_j
- At each step, the feature X^* such that

$$X^* = \arg \max_{X_i \notin \mathcal{S}_c} \left\{ \min_{X_j \in \mathcal{S}_c} \mathbb{I}(X_i, C|X_j) \right\}$$

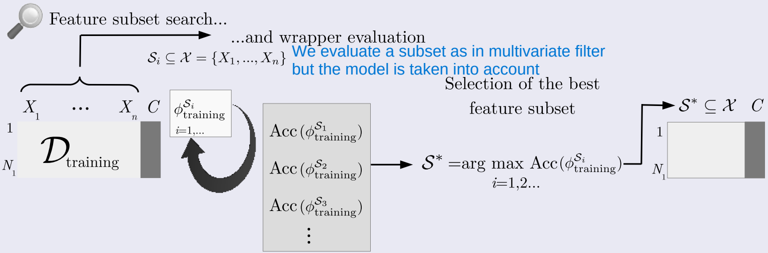
is added to the current subset $\mathcal{S}_c$ containing the selected features

Outline

Wrapper Approaches

Wrapper methods (John et al., 1994; Langley and Sage, 1994) evaluate each possible subset of features with a criterion consisting of the estimated performance of the classifier built with this subset of features

Schema of a wrapper approach

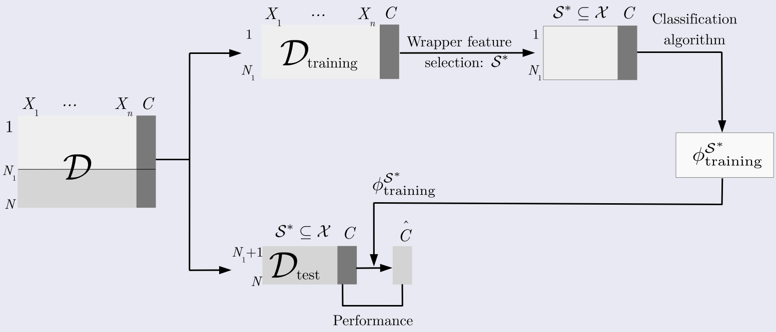


We evaluate a subset as in multivariate filter but the model is taken into account

Wrapper Approaches

More computationally demanding than the filter approach

Evaluation of the classification model output by a wrapper FSS



Wrapper Approaches

Heuristics search strategies

Deterministic heuristics:	
Sequential feature selection	Fu (1968)
Sequential forward feature selection	Fu (1968)
Sequential backward elimination	Marill and Green (1963)
Greedy hill climbing	John et al. (1994)
Best first	Xu et al. (1988)
Plus-L-Minus-r algorithm	Stearns (1976)
Floating search selection	Pudil et al. (1994)
Tabu search	Zhang and Sun (2002)
Branch and bound	Lawler and Wood (1966)
Non-deterministic heuristics:	
Single-solution metaheuristics:	
Simulated annealing	Doak (1992)
Las Vegas algorithm	Liu and Motoda (1998)
Greedy randomized adaptive search procedure	Bermejo et al. (2011)
Variable neighborhood search	Garcia-Torres et al. (2005)
Population-based metaheuristics:	
Scatter search	Garcia-Lopez et al. (2006)
Ant colony optimization	Al-An (2005)
Particle swarm optimization	Lin et al. (2008)
Evolutionary algorithms:	
Genetic algorithms	Siedlecki and Sklansky (1989)
Estimation of distribution algorithms	Inza et al. (2000)
Differential evolution	Khushaba et al. (2008)
Genetic programming	Muni et al. (2004)
Evolution strategies	Vatolkin et al. (2009)

Wrapper Approaches

Heuristics search strategies. Variable neighborhood search algorithm

Algorithm 2: The variable neighborhood search algorithm

Input : A set of neighborhood structures $\mathfrak{N} = \{N_1, N_2, ..., N_{max}\}$ for shaking
Output: Best solution found

```
repeat
    k = 1
    repeat
        Shaking: pick a random solution s' from the kth neighborhood N_k(s) of s
        Local search: apply local search to s' to get s''
        if f(s'') > f(s) then s = s''
        Continue to search with N_1; k = 1
        else k = k + 1
    until k = max
until Stopping criterion is satisfied
```

Wrapper Approaches

Heuristics search strategies. Evolutionary algorithms

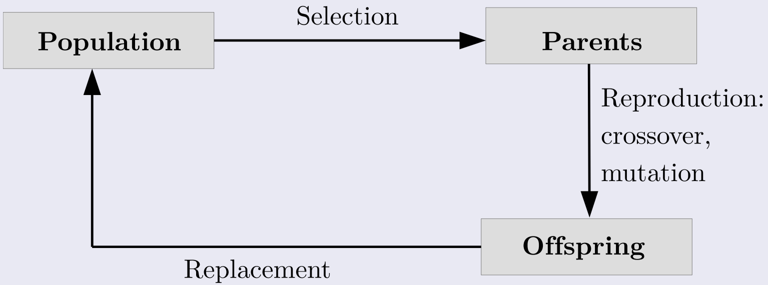
Algorithm 3: An evolutionary algorithm

Input : Generate the initial population, $Pop(0)$
Output: Best individual found

while *Stopping criterion($Pop(t)$) is not met* **do**
    Evaluate($Pop(t)$)
    $Pop'(t)$ = Selection($Pop(t)$)
    $Pop'(t)$ = Reproduction($Pop'(t)$); Evaluate($Pop'(t)$)
    $Pop(t+1)$ = Replace($Pop(t)$, $Pop'(t)$)
    $t = t + 1$
endwhile

Wrapper Approaches

Heuristics search strategies. Genetic algorithms



Basic scheme of a generation in a genetic algorithm

Outline

Hybrid Feature Selection

Hybrid feature selection methods combine filter and wrapper approaches, especially when the initial number of features is so large that wrapper methods cannot be used on computational grounds

Minimal-redundancy-maximal-relevance (Peng et al. 2005)

1. The subset is selected in a multivariate filter approach

$$\mathcal{S}^* = \arg\max_{\mathcal{S}\subseteq\mathcal{X}} \Phi_{(r,R)}(\mathcal{S}, C) = \arg\max_{\mathcal{S}\subseteq\mathcal{X}} \left(R(\mathcal{S}, C) - r(\mathcal{S}, C)\right)$$

   where $R(\mathcal{S}, C) = \frac{1}{|\mathcal{S}|}\sum_{X_i\in\mathcal{S}} \mathbb{I}(X_i, C)$ denotes the relevance and $r(\mathcal{S}, C) = \frac{1}{|\mathcal{S}|^2}\sum_{X_i,X_j\in\mathcal{S}} \mathbb{I}(X_i, X_j)$ the redundancy

2. A wrapper approach is applied to this subset $\mathcal{S}^*$

Outline

Feature subset selection methods

- Necessary in nowadays machine learning
- Filter approaches: univariate and multivariate
- Wrapper approaches: need the use of heuristics search algorithms
- Hybrid methods: combine filter (first) and wrapper (second)

FEATURE SUBSET SELECTION

Pedro Larrañaga, Concha Bielza

Computational Intelligence Group
Artificial Intelligence Department
Universidad Politécnica de Madrid

Machine Learning
Master in Data Science + Master in HMDA