Filter methods for feature selection. A comparative study *

Noelia Sánchez-Maroño, Amparo Alonso-Betanzos, and María Tombilla-Sanromán

University of A Coruña, Department of Computer Science, 15071 A Coruña, Spain
nsanchez@udc.es, ciamparo@udc.es, infmts00@ucv.udc.es

Abstract. Adequate selection of features may improve accuracy and efficiency of classifier methods. There are two main approaches for feature selection: wrapper methods, in which the features are selected using the classifier, and filter methods, in which the selection of features is independent of the classifier used. Although the wrapper approach may obtain better performances, it requires greater computational resources. For this reason, lately a new paradigm, hybrid approach, that combines both filter and wrapper methods has emerged. One of its problems is to select the filter method that gives the best relevance index for each case, and this is not an easy to solve question. Different approaches to relevance evaluation lead to a large number of indices for ranking and selection. In this paper, several filter methods are applied over artificial data sets with different number of relevant features, level of noise in the output, interaction between features and increasing number of samples. The results obtained for the four filters studied (ReliefF, Correlation-based Feature Selection, Fast Correlated Based Filter and INTERACT) are compared and discussed. The final aim of this study is to select a filter to construct a hybrid method for feature selection.

1 Introduction

Enhancement of generalization, i.e., the performance of the learning algorithm over the test set, often motivates feature selection, which consists on detecting the relevant features and discarding the irrelevant features. Feature selection has several advantages [**?**], such as:

- improving the performance of the machine learning algorithm.
- data understanding, gaining knowledge about the process and perhaps helping to visualize it.
- data reduction, limiting storage requirements and perhaps helping in reducing costs.
- simplicity, possibility of using simpler models and gaining speed.

* This work has been funded in part by Project PGIDT05TIC10502PR of the Xunta de Galicia and TIN2006-02402 of the Ministerio de Educación y Ciencia, Spain (partially supported by the European Union ERDF).

There are two main models that deal with feature selection: filter methods and wrapper methods [**?**]. While wrapper models involve optimizing a predictor as part of the selection process, filter models rely on the general characteristics of the training data to select features with independence of any predictor. Wrapper models tend to give better results but filter methods are usually computationally less expensive than wrappers. So, in those cases in which the number of features is very large, filter methods are indispensable to obtain a reduced set of features that then can be treated by other more expensive feature selection methods. In fact, this is the basis of the most recent hybrid algorithms for feature selection, that try to take the strong points of both previous approaches [**?**,**?**]. One of the problems that is needed to be faced is which filter gives the best relevance index for each case, and this is not an easy to solve question. Different approaches to relevance evaluation lead to a large number of indices for ranking and selection [**?**]. In this paper, several filter methods are applied over several synthetic problems to test its effectiveness under different situations: increasing number of relevant features and samples, noisy output and interaction between features.

2 The Filter methods used

As described above, filter methods carry out the feature selection process as a pre-processing step with no induction algorithm. The general characteristics of the training data are used to select features (for example, distances between classes or statistical dependencies). This model is faster than the wrapper approach and results in a better generalization because it acts independently of the induction algorithm. However, it tends to select subsets with a high number of features (even all the features) and so a threshold is required to choose a subset.

In this paper, a comparison over several artificial problems is carried out. The filter methods selected are the following:

2.1 Relief

The original RELIEF algorithm [**?**] estimates the quality of attributes according to how well their values distinguish between instances that are near to each other. For this purpose, given a randomly selected instance, $\mathbf{x}_i = \{x_{1i}, x_{2i}, , x_{ni}\}$, RELIEF searches for its two nearest neighbours: one from the same class, called nearest hit H, and the other from a different class, called nearest miss M. It then updates the quality estimate for all the features, depending on the values for $\mathbf{x}_i$, M, and H. The original RELIEF can deal with discrete and continuous features but is limited to two-class problems. An extension, ReliefF [**?**], not only deals with multiclass problems but is also more robust and capable of dealing with incomplete and noisy data. ReliefF was subsequently adapted for continuous class (regression) problems, resulting in the RReliefF algorithm [**?**]. The Relief family of methods are specially attractive because they may be applied in all situations, have low bias, include interaction among features and may capture local dependencies that other methods miss.

2.2 Correlation-based Feature Selection, CFS

Correlation based Feature Selection (CFS) is a simple filter algorithm that ranks feature subsets according to a correlation based heuristic evaluation function [**?**]. The bias of the evaluation function is toward subsets that contain features that are highly correlated with the class and uncorrelated with each other. Irrelevant features should be ignored because they will have low correlation with the class. Redundant features should be screened out as they will be highly correlated with one or more of the remaining features. The acceptance of a feature will depend on the extent to which it predicts classes in areas of the instance space not already predicted by other features. CFS's feature subset evaluation function is:

$$M_S = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}},$$

where M_S is the heuristic "merit" of a feature subset S containing k features, $\overline{r_{cf}}$ is the mean feature-class correlation ($f \in S$) and $\overline{r_{ff}}$ is the average feature-feature intercorrelation. The numerator of this equation can be thought of as providing an indication of how predictive of the class a set of features is; and the denominator of how much redundancy there is among the features.

2.3 Fast correlated based filter-FCBF

The fast correlated-based filter (FCBF) method [**?**] is based on symmetrical uncertainty (SU) [**?**], which is defined as the ratio between the information gain (IG) and the entropy (H) of two features, x and y:

$$SU(x, y) = 2\frac{IG(x/y)}{H(x) + H(y)}, \tag{1}$$

where the information gain is defined as:

$$IG(x/y) = H(y) + H(x) - H(x, y),$$

being $H(x)$ and $H(x, y)$ the entropy and joint entropy, respectively. This method was designed for high-dimensionality data and has been shown to be effective in removing both irrelevant and redundant features. However, it fails to take into consideration the interaction between features.

2.4 INTERACT

The INTERACT algorithm [**?**] uses the same goodness measure as FCBF filter, i.e., SU in (**??**), but it also includes the consistency contribution (c-contribution). C-contribution of a feature is an indicator about how significantly the elimination of that feature will affect consistency. The algorithm consists of two major parts. In the first part, the features are ranked in descending order based on their SU values. In the second part, features are evaluated one by one starting from the end of the ranked feature list. If c-contribution of a feature is less than an established threshold, the feature is removed, otherwise it is selected. The authors stated in [**?**] that INTERACT can thus handle feature interaction, and efficiently selects relevant features.

3 Synthetic datasets

In order to determine the effectiveness of each filter at different situations a synthetic dataset was generated. The idea of this synthetic problem was suggested by the work in [**?**]. Consider m pairs of training samples:

$$\{\mathbf{x}_1, y_1\}, \{\mathbf{x}_2, y_2\}, \ldots, \{\mathbf{x}_m, y_m\},$$

where $\mathbf{x}_i = [x_{1i}, x_{2i}, \ldots, x_{ni}]^T$ is a $n-$dimensional feature vector representing the i^{th} training sample, and $y_i \in [-1, 1]$ is the class label of $\mathbf{x}_i$. The number of samples considered was 400. In the original problem [**?**], the features were uniformly distributed in the rank $(0, 1)$, however most of the filter methods required discrete features in order to work properly. Although discretization methods, such Fayyad and Irani's MDL method [**?**], can be applied as a preprocessing step to overcome this problem, it was observed that this process discards some features (for example, by assigning a unique value to them) and therefore different filter methods can lead to the same set of features. Then, and in order to avoid that, a set of 200 discrete features uniformly distributed in the rank $[0, 5]$ was used.

Different experiments were carried out based on that set of input features to test the behavior of the filters. Four different situations were considered: a)different number of relevant features, b) adding noise to the output, c) interaction between features and d) different ratios between the number of samples and the number of features.

To tackle with the first two situations, the desired output was estimated as:

$$\hat{y}_i = \sum_{j=1}^{R} a_j x_{ji}, \tag{2}$$

where R is the number of relevant features considered from the whole set and a_j is randomly fixed to -1 or 1. In order to achieve a perfect balanced binary classification problem, the desired output $\mathbf{y}$ was evaluated as:

$$y_i = \begin{cases} \text{-1} & \text{if } y_i < \text{median}(\hat{\mathbf{y}}) \ , \\ 1 & \text{if } y_i >= \text{median}(\hat{\mathbf{y}}) \ . \end{cases} \tag{3}$$

The number of relevant features can be 10, 20, 30 or 40, and the percentage of noise varies from 0% to 20%. Note that, as a binary classification problem, adding noise to the output means assigning some outputs to the wrong class.

The equation in (**??**) does not consider interaction between variables. Therefore, the following variations were taken into account:

$$y_i = \sum_{j=1}^{R} a_j x_{ji} + b_1(x_{f_1} x_{f_2}) + b_2(x_{f_2} x_{f_3}), \tag{4}$$

$$y_i = \sum_{j=1}^{R} a_j x_{ji} + b_1(x_{f_1} x_{f_2} x_{f_3}) \tag{5}$$

In these cases, R changes from 10 to 20 and the desired output is without noise. Different problems were proposed based on equations (**??**) and (**??**), considering that the features x_f can be in both the set of R relevant features and in the set of irrelevant features. For example, one of the considered problems was:

$$y_i = \sum_{j=1}^{10} a_j x_{ji} + b_1(x_3 x_4) + b_2(x_4 x_6),$$

that involves the features $\{x_3, x_4, x_6\}$, and all of them are relevant by its own because they are included in the first term of the equation (**??**). The goal of this kind of problems was to test if the relevance of those features varies. On the other hand, the following problem adds new relevant features:

$$y_i = \sum_{j=1}^{10} a_j x_{ji} + b_1(x_{13} x_{14}) + b_2(x_{14} x_{15}).$$

Table 1. Results for ReliefF. R indicates the number of relevant features and N is the percentage of noisy outputs

R	N (%)	10-first selected features	% of success
10	0	**4**, **7**, 71, 39, 173, **1**, **3**, 176, **6**, 190, 115 ...	50
	5	**7**, **3**, **4**, **8**, 71, 109, 139, 190, 161, 176...	40
	10	**8**, 173, 121, 169, 71, 137, **7**, 148, 161, 176...	20
	15	148, 72, 49, 11, 118, 135, 158, 27, 38, **10**...	10
	20	148, 79, 76, 133, 158, **1**, 44, 112, 34, 72...	10
20	0	**10**, **6**, 199, 112, **17**, **18**, **9**, 39, **19**, 91 ...	40
	5	**18**, 112, 199, **10**, **1**, 155, 59, **17**, **14**, 102...	30
	10	**1**, **18**, 112, **2**, 115, 40, 160, 63, 184, 48...	25
	15	68, 111, 176, **5**, 39, 80, 131, 136, 199, 63...	20
	20	179, 49, 92, 150, 177, 119, 182, 50, 83, 167...	5
30	0	**1**, 188, 19, 182, 155, 17, 191, 31, 42, 104...	23, 3
	5	**17**, **1**, 73, 191, 93, 184, 32, 19, 182, 31...	23, 3
	10	191, **17**, 182, 104, **1**, 174, 138, 144, 75, 107...	20
	15	**13**, 75, 177, 71, 44, **19**, **8**, **20**, 100, 93...	16.6
	20	91,146,100,162,31,84,184,93,168,150...	16, 6
40	0	**32**, 44, **17**, 107, 90, 128, **33**, **35**, 74, **10**...	25
	5	**32**, 74, 125, 85, 44, **35**, 43, 90, 101, **17**...	25
	10	**32**, 68, 125, **17**, 44, **29**, **35**, **25**, 43, **8**...	27
	15	45, 152, 44, 99, 111, **32**, 123, 61, 134, **17**...	22, 5
	20	41, 45, **40**, 83, 93, 134, **28**, 59, 49, 70...	12, 5

4 Experimental results

The filters used for this study, except INTERACT, are available at the data mining software WEKA [**?**]. INTERACT can be downloaded from [**?**]. Tables **??**, **??** and **??** show the results achieved when different number of relevant features and degrees of noise are taken under consideration with ReliefF, FCBF and CFS, respectively. INTERACT was not considered because of its similarity to FCBF. In those tables, R indicates the number of relevant features and N the percentage of noise considered. The features are consecutively labeled from 1 to 200 and the relevant features are placed at the beginning. So, if R is equal to 10, the relevant features are $\{x_1, x_2, \ldots, x_{10}\}$ and all the others are irrelevant. The percentage of success shown in the last column of Tables **??**-**??** indicates the percentage of correctly selected features between the R relevant ones; for example, in the first row of Table **??** the features 1 to 10 should be detected as relevant, however only 5 of them are included $\{4, 7, 1, 3, 6\}$.

Table 2. Results for FCBF. R indicates the number of relevant features and N is the percentage of noisy outputs

R	N (%)	10-first selected features	%success
10	0	**8**, **3**, **4**, **7**, **2**, **10**, **6**, **1**, **9**, **5** ...	100
	5	**2**, **4**, **3**, **8**, **7**, **10**, **1**, **6**, **9**, 25...	90
	10	**2**, **8**, 94, **3**, **6**, **1**, 38, 71, 145, 7 ...	60
	15	145, 117, 72,11, **8**, **10**, **2**, **6**, 168, 94...	40
	20	44, 71, 72, 145, **2**, 93, 175, 22, 26, 47...	10
20	0	**8**, **17**, **14**, **11**, **6**, **20**, 191, **4**, 39, 172...	75
	5	**8**, **17**, **20**, **14**, 172, **16**, **11**, **18**, 38, **3**...	70
	10	**17**, 38, **11**, **8**, **1**, **18**, 172, 104, **3**, **15**...	65
	15	**17**, **1**, 39, **2**, 115, 68, **10**, **16**, 93, 29...	40
	20	**8**, **16**, **17**, 127, 120, 130, **1**, 35, 200, 72...	25
30	0	**26**, **14**, **8**, **15**, **30**, **11**, **28**, **17**, **1**, **19**...	60
	5	**14**, **17**, **30**, **11**, **26**, **8**, **15**, **20**, 119, **21**...	50
	10	147,**8**, **17**, **26**, 38, 125, 104, **3**, **30**, 138...	43
	15	145, **21**, **8**, 189, **18**, 83, **17**, **23**, 105, **3**...	30
	20	145, **17**, 93, 53, **21**, **9**, 191, 71, 150, 195...	17
40	0	**26**, 34, 35, **17**, **14**, **19**, 33, 182, **30**, 134...	52
	5	**26**, **35**, **11**, 115, **19**, **28**, **14**, **34**, **30**, **33**...	47
	10	**35**, **28**, 182, 149, **30**, 1432, 102, **17**, **19**...	40
	15	**35**, 149, 144, **32**, 69, 153, 167, 182, **11**, **20**...	35
	20	172,193,83,64,169,125,93,87,150,69...	25

As it is stated in the literature, ReliefF and FCBF return the whole set of features ranked according to its "internal" measures and they require thresholds to select a subset. Tables **??**-**??** show the first positions of the ranked lists. It is important to remark the variability observed in the higher values of the ranked list from one problem to another. Using ReliefF, the feature placed at the first position gets a value equal to $0,0375$ when solving the simplest problem,

R=10-N=0, however this first value changes to $0,0278$ in the most difficult case, R=40-N=20. The differences are even more drastic using FCBF, being the first values $0,0375$ and $0,0169$, respectively. This fact remarks the difficulty of finding a proper threshold for each problem.

Table 3. Results for CFS filter. S is the subset of features that returns, $|S|$ is its cardinality and %R stands for the percentage of relevant features included in S.

R	N (%)	Relevant Features ∈ S	$\|S\|$	%R
10	0	[1-10]	14	100
	5	[1-10]	12	100
	10	[1-10]	35	100
	15	1,2,3,6,8,10	35	60
	20	2,8	37	20
20	0	[3-11],[14-20]	22	80
	5	1,[3-8],10,11,[11-20]	26	80
	10	1,2,3,[5-11],[14-18],20	25	80
	15	[1-5],[8-11],[13-17],19,20	39	80
	20	1, 2, 8, 13, 16, 17, 20	56	35
30	0	1, 4, 8, 9, 11, 13, 14, 15, 17, 20, 21, 23, 25, 26,28,30	22	50
	5	1, 3, 8, 11, 13, 14, 15, 17, 20, 21, 23, 26, 28, 30	27	47
	10	1, 3, 8, 14, 15, 16, 17, 18, 20, 21, 23, 26, 30	30	43,3
	15	3, 4, 8, 10, 13, 17, 18, 21, 23, 26, 27	36	36,3
	20	9, 17, 21, 25, 27	36	16,6
40	0	6, 8, 10, 11, 13, 14, 15, 17, 19, 20, 21, 23, 25, 26, 28, 30, [32- 35], 37	37	52,5
	5	4, 8, 10, 11, 14, 15, 17, 19, 20, 21, 23, 25, 26, 28, 29, 30, [32-35], 37	43	52,5
	10	10, 11, 14, 17, 19, 21, 23, 25, 26, 28, 30, 31, 32, 33, 35, 36	32	40
	15	5, 10, 11, 14, 17, 20, 21, 24, 26, 28, 29, 30, 31, 32, 35, 38	46	40
	20	1 , 11, 12, 13, 18, 27, 28, 29, 32, 35	43	25

On the other hand, CFS returns a subset of features. In Table **??** can be observed that the features selected are the correct ones for the simplest cases, R=10, N=0, 5 and 10. However, the size of the selected subset increases considerably when the level of noise increases. Note that the last column of Table **??** shows the percentage of relevant features included in the subset S returned by the filter method. This performance measure is different from the one used for ReliefF and FCBF and therefore a fair comparison is not directly possible. However, it can be stated that both FCBF and CFS exhibit good performance results in the examples considered.

Tables **??** and **??** show the results obtained with interaction problems (see equations **??**-**??**). The first columns of both tables present the different interaction problems considered. Similarly to a_j in (**??**), the values of b_1 and b_2 in equations (**??**)-(**??**) were equal to $0,2$ or $-0,2$, randomly chosen, unless for test number 7 in which their value was $0,4$. INTERACT was selected instead of FCBF because it is better suited for this type of problems [**?**]. Table **??** presents

the complete set of features obtained for INTERACT and CFS. It can be noticed that, in test 2, both methods are able to detect the added features $\{x_{13}, x_{14}, x_{15}\}$. Besides, CFS seems to detect the interaction better than INTERACT (tests 5, 6 and 8), although it also returns a higher number of features. Test 7 suggests that the interaction term should have a high value in order to be considered for INTERACT.

Table 4. Results obtained with Interact and CFS using different problems of interaction between features based on equations (**??**) and (**??**). f_1, f_2 and f_3 indicate the indexes of the selected features. Tests shown over the double line use 10 relevant features, while those below it use 20 relevant features.

Test	Eq.	f_1	f_2	f_3	INTERACT	CFS
1	(**??**)	1	7	24	2,3,4,5,10,**24**	[1-5],8,9,10,12,**24**,35,180
2	(**??**)	13	14	15	$4, 6, 8, \mathbf{13}, \mathbf{14}, \mathbf{15}$	$1, 2, 4, [6-9], \mathbf{13}, \mathbf{14}, \mathbf{15}, 24, 123, 129, 138, 149$
4	(**??**)	56	7	64	1,2,3,4,5,10	[1-6],8,9,10,**64**,149
5	(**??**)	90	100	-	3,6,8,11,14,17,18	[1-11],[14-21],26,39,44,60,**100** ,104,112,164,172,180,191
6	(**??**)	100	200	150	4,6,8,11,17,19	[1-11],[13-20],38,44,**100**, 112,120,123,144,172,180,185,191,197,**200**
7	(**??**)	90	100	-	6,8,17,19,100,112	[1-4],6,7,8,10,11,[14-20],34,38,39,51,60, **100**,112,113,120,123,164,168,172,180,189,191
8	(**??**)	100	5	200	3,6,8,11,14,17,18	[1-4],[6-11],[14-21],26,39,44,60,**100** ,104,112,164,172,180,191

Table 5. Results obtained ReliefF using different problems of interaction between features based on equations (**??**) and (**??**). F stands for the feature index and P stands for its position at the ranked list. Tests shown over the double line use 10 relevant features, while those below it use 20 relevant features.

Test	Eq.	f_1	f_2	f_3	Interaction						No Interaction					
					F	P	F	P	F	P	F	P	F	P	F	P
1	(**??**)	1	7	24	1	2	7	200	24	8	1	6	7	3	24	193
2	(**??**)	13	14	15	13	4	14	2	15	5	13	164	14	74	15	89
3	(**??**)	2	4	6	2	1	4	2	6	6	2	46	4	1	6	9
4	(**??**)	56	7	64	56	55	7	145	64	26	7	3	56	119	64	140
7	(**??**)	90	100	-	90	165	100	21	-	-	90	147	100	185	-	-

As ReliefF returns a ranked list of features, the goal was to look for differences between both lists of ranked features, with and without interaction. This comparison is shown in Table **??**. Similarly to the results of INTERACT and CFS, test 2 gets very good results, with the involved variables $\{x_{13}, x_{14}, x_{15}\}$ getting very high positions at the list, indicating their relevance. However, test 1 shows some poor results, because feature 7, included in three terms of the equation (**??**), gets the last position. Test 3 and 4 also denote a proper selection of the filter. Regarding the tests with R=20, some slight differences were observed in the lists, however they are not remarkable and only test 7 of table

?? is included in Table **??** to denote the increased value of feature 100, although feature 90 remains in a similar position.

Finally, table **??** includes the results obtained using different number of samples. ReliefF and FCBF returned a very different ranked list, and so a different threshold should be used for each. For example, fixing a threshold equal to 0.03 for FCBF will lead to 53 features in the first case ($R = 10$ and $m = 100$), while it will return an empty list with $R = 40$ and $m = 800$. Therefore, the $\%success$ was used instead of the threshold. The $\%success$ indicates the number of correct relevant features placed at the first R positions of the list. $R \subset S$ pinpoints the percentage of relevant features included in S and it is used for CFS and INTERACT. As it can be seen, CFS and FCBF present good performance results, and that FCBF gets poorer results than CFS when 40 relevant features are used. INTERACT exhibits a poor performance, but on the other hand it returns a very small set of features.

Table 6. Results with different number of samples. $\%success$ indicates the number of relevant features placed at the first R features of the list. S is the subset of features that is returned by INTERACT and CFS, $|S|$ is its cardinality and %R stands for the percentage of relevant features included in S.

R	Number	ReliefF	FCBF	INTERACT		CFS	
	of samples	%success	%success	$\|S\|$	R ⊂ S (%)	$\|S\|$	R ⊂ S (%)
10	100	20	30	5	20	14	50
	200	20	50	4	10	18	80
	400	40	100	8	80	9	90
	800	60	100	10	100	10	100
20	100	25	20	5	0	15	20
	200	20	50	6	25	22	50
	400	30	30	7	15	20	35
	800	60	85	14	65	20	90
30	100	13,3	26,7	4	3,3	9	6,7
	200	13, 3	40	5	6,7	20	30
	400	3	70	6	20	35	73,3
	800	40	62,5	13	40	21	66,7
40	100	25	25	5	5	13	10
	20	15	32,5	5	5	33	32,5
	400	27,5	90	6	15	33	55
	800	32,5	90	7	17,5	34	30

5 Conclusions

In this paper, four filter methods for feature selection are applied over a synthetic data set aimed at studying the performance of the methods regarding the number of features and samples, the level of noise and the interaction between features . All the methods work better with a reduced set of features, and the results get worse when the number of features increases. However, in the case of the CFS and FCBF, performance in the case of 40 relevant features is much better than for ReliefF. Regarding the level of noise, again all methods reduce their performance

when the level of noise increases, but FCBF and CFS are the ones in which this worsening is smoother, specially for CFS. Regarding the performance of the filter methods when considering interaction between features, ReliefF has not a solid behaviour, and CFS is again the one in which the results are more promising. Finally, CFS and FCBF exhibits an acceptable performance with a reduced set of samples. So, in summary, CFS is the filter method more adequate for our purposes, because it is the one that obtains better performance and maintains a smoother decrement of it when the number of features and the level of noise increment. Besides, it presents an adequate detection of interaction between features, a complicated aspect for filters, as can be seen in the results presented on the previous section. It can be argued that CFS is the method that has a higher cardinality, that is, detects more features than the real relevant ones. Although this is true, the number of features in excess is not high at all (around 7 more than needed, when noise is around 5-10%) and this is perfectly assumable when the filter is a previous step for eliminating irrelevant features, that will be followed by a wrapper method.

References

1. Guyon, I., Gunn, S., Nikravesh, M., Zadeh, L.: Feature Extraction. Foundations and Applications. Springer (2006)
2. Kohavi, R., John, G.: Wrappers for feature subset selection. Artificial Intelligence Journal, Special issue on relevance **97**(1-2) (1997) 273–324
3. Liu, H., Dougherty, E., Gy, J.D., Torkkola, K., Tuv, E., Peng, H., Ding, C., Long, F., Berens, M., Yu, L., G.Forman: Evolving feature selection. IEEE Intelligent systems **20** (2005) 64–76
4. Kira, K., Rendell, L.: A practical approach to feature selection. In: Proceedings of the Ninth International Conference on Machine Learning. (1992) 249–256
5. Kononenko, I.: Estimating attributes: Analysis and extensions of RELIEF. In: European Conference on Machine Learning. (1994) 171–182
6. Robnik-Sikonja, M., Kononenko, I.: Theoretical and empirical analysis of ReliefF and RReliefF. Machine Learning **53** (2003) 23–69
7. Hall, M.A.: Correlation-based Feature Selection for Machine Learning. PhD thesis, University of Waikato, Hamilton, New Zealand (1999)
8. Yu, L., Liu, H.: Feature selection for high-dimensional data: A fast correlation-based filter solution. In: Proceedings of The Twentieth International Conference on Machine Learning, ICML. (2003) 856–863
9. Press, W.H., Flannery, B.P., Teukolsky, S.A., Vetterling, W.T.: Numerical recipes in C. Cambridge University Press, Cambridge (1988)
10. Zhao, Z., Liu, H.: Searching for interacting features. In: Proceedings of International Joint Conference on Artificial Intelligence, IJCAI. (2007) 1156–1161
11. Quevedo, J.R., Bahamonde, A., Luaces, O.: A simple and efficient method for variable ranking according to their usefulness for learning. Journal Computational Statistics and Data Analysis (2007) (in press)
12. Fayyad, U.M., Irani, K.B.: On the handling of continuous-valued attributes in decision tree generation. Machine Learning **8** (1992) 87–102
13. WEKA Machine Learning Project `http://www.cs.waikato.ac.nz/~ml/` Last access: September 2007.
14. Liu, H.:Searching for interacting features `http://www.public.asu.edu/~huanliu/INTERACT/INTERACTsoftware.html` Last access: September 2007.