

Infosel++: Information Based Feature Selection C++ Library

A. Kachel¹, J. Biesiada^{1,3}, M. Blachnik¹ and W. Duch²

¹ Silesian University of Technology, Electrotechnology Department, Katowice, Krasińskiego 8, Poland; Jacek.Biesiada@polsl.pl

² Department of Informatics, Nicolaus Copernicus University, Grudziądzka 5, Toruń, Poland; Google: W. Duch

³ Division of Biomedical Informatics, Children's Hospital Research Foundation, 3333 Burnet Ave., Cincinnati, Ohio 45229

Abstract. A large package of algorithms for feature ranking and selection has been developed. Infosel++, Information Based Feature Selection C++ Library, is a collection of classes and utilities based on probability estimation that can help developers of machine learning methods in rapid interfacing of feature selection algorithms, aid users in selecting an appropriate algorithm for a given task (embed feature selection in machine learning task), and aid researchers in developing new algorithms, especially hybrid algorithms for feature selection. A few examples of such possibilities are presented.

1 Introduction

Feature selection and ranking is an essential step in all data mining projects [1]. In bioinformatics, text analysis, object recognition or in modeling of complex technological processes large number of features is generated, and from a specific point of view, frequently related to recognition of some target concepts, only a small subset of features is usually relevant. Moreover, strategy based on systematic construction of many types of features followed by selection appears to be very powerful in finding simple models of data [2]. With limited amount of available data many spurious models in highly dimensional feature spaces may be created due to the accidental correlations between the target concept and various ways of partitioning the data, making these solutions worthless. To deal with such problems dimensionality of the feature space has to be reduced first. This may be done by ranking these features and selecting the most important ones, selecting a subset of relevant features or by combining (aggregating) subsets of features to create new, more informative features.

According to fundamental results in computational learning theory no single method is the best in all situations, and no single feature selection algorithm is the best for all data and all tasks. Many feature selection and feature ranking methods have been proposed in the literature [3, 1, 4, 5]. Although numerous libraries of learning methods have been created (as detailed in the next section) libraries of feature selection algorithms are not so popular. In this contribution we present InfoSel++, a library based on standard as well as novel algorithms. These algorithms may be used in several ways: for ranking, feature selection based on filters, or as a combination of filters and wrappers

called frappers [6]. Ranking of features neglects their possible interactions, assigning relevancy to individual features and introducing partial order among features. Many measures of relevancy based on statistics and information theory suitable for ranking methods have been implemented. Wrappers use the results of predictors to evaluate the usefulness of features, but need a learning algorithm to control feature selection process. Wrapper methods employ statistical re-sampling techniques (such as cross-validation) using specific learning algorithms to estimate the accuracy of feature subsets. This approach has proved useful, but may be computationally very demanding because the learning algorithm is called repeatedly. Model selection techniques should be used to avoid over-fitting. For this reason wrappers do not scale well to large datasets containing many features. Filter methods, on the other hand, operate independently of any learning algorithm, searching for potentially useful dependencies between target task and distribution of feature values. Typically they attempt to rank features according to a relevancy score, but may also be used for selection of subsets of features [6].

In the next section various projects where feature selection has been prominent are reviewed. The third section describes our InfoSel library, section 4 contains a few results and comparisons, and the final section a brief discussion.

2 Related Work

Several large-scale efforts that implement libraries of machine learning algorithms have been undertaken in the past, and many of them include special modules for feature selection. Some are designed for general tasks, and some are specialized in such areas as microarray gene selection analysis [3]. Large projects involving feature selection as a part of a bigger system are listed first (MLC++, Weka, GhostMiner, Matlab and R packages, ToolDiag), followed by smaller and more specialized projects (RankGene, Feature Selection Toolbox).

1. **The Machine Learning in C++ library** (MLC++) [7] for supervised data mining problems has been developed at Stanford University. It includes decision trees and decision tables, Naive Bayes, instance based algorithms and many other machine learning techniques. It provides an implementation of wrapping approach for feature selection utilizing best first search, forward and backward selection methods.

2. **Weka** [8] is a popular large data mining environment developed at Waikato University, New Zealand, that is still being rapidly developed and used as a part of newer packages, such as RapidMiner [9]. Among many computational tools implemented, it contains about 15 attribute and subset evaluator methods, extended by 10 search algorithms for feature selection. Weka's feature selection algorithms are grouped into two subcategories: *feature evaluators* and *search methods*. The former group is used for evaluation of relevance of single features or feature subsets, usually by estimating various ranking coefficients, including: information gain, gain ratio, Chi squared statistic, oneR tree index, significance index, symmetrical uncertainty index. The latter category is an implementation of different optimal and sub-optimal search methods that use feature evaluators as a cost function.

3. **GhostMiner** is a commercial data mining tool distributed by Fujitsu. It has several classifiers, including various versions of SVM, decision tree, kNN with feature

weighting and instance selection methods, incremental neural network, and a Feature Space Mapping neuro-fuzzy network. It also implements a few most effective feature selection methods like forward and backward selection, ranking based on various coefficients, and estimation of feature subset quality by wrapping with any classifier.

4. **PRTool and Spider** are Matlab toolboxes designed for data mining, neural networks and machine learning. **PRTool** has been developed by the Pattern Recognition Group at Delft University and is freely distributed. This toolbox implements various pattern recognition algorithms (kNN, decision trees, Parzen classifier, etc) that may be combined with feature selection methods (ranking of individual features, various search methods). **Spider** (Max Planck Institute of Biological Cybernetics, Germany) is delivered under the GNU license. It allows for an easy creation of two-staged data mining tasks. Available feature selection methods include mutual information filters, Fisher/Correlation score, greedy selection algorithm, L_0 zero-norm minimization, primal zero-norm based feature selection, feature scaling using SVMs, non-linear feature elimination, and multi-class feature selection using spectral clustering. These methods may be combined with various classification and regression algorithms.

5. **R-project packages** (GNU license), similarly to Matlab toolboxes, are designed to solve dedicated computational problems. Feature selection using R is available through the *FSelector* package. It includes all feature selection algorithms implemented in Weka. Additional smaller packages are also available. *PenalizedSVM*, provides the smoothly clipped absolute deviation (SCAD) and L_1 -norm penalty functions for SVM based feature selection. The *SDA* package (Shrinkage Discriminant Analysis and Feature Selection) offers a classifier that can be trained using Stein-type shrinkage estimators where features are ranked using correlation-adjusted t -scores. *Bioconductor* is an open source software project for the analysis of genomic data, gene selection and association analysis. Finally, the *predmixcor* package creates classification rules based on Bayesian mixture models with feature selection bias corrected approach.

6. **ToolDiag** is another general data mining and pattern recognition tool. It includes several decision making algorithms such as artificial neural networks (MLP, RBF, LVQ), kNN, linear, quadratic and Parzen classifiers. Also available are basic preprocessing and statistical analysis methods. Model estimation approaches include resubstitution error, holdout, cross-validation, bootstrap and leave-one-out. Tools for feature selection include 5 search algorithms (best features, sequential forward and sequential backward selection, branch and bound, exhaustive search) that can be combined with three groups of selection criteria: estimated minimal error probability, inter-class distance (using different distance matrices), and probabilistic distances (Chernoff, Bhattacharyya, Jeffreys-Matusita, Patrick-Fisher and Mahalanobis distance, KL divergence).

7. **Feature Selection Toolbox** [10] is an advanced tool developed by Petr Somol and his group. The software includes some classical and new methods of dimensionality reduction, classification and data representation. The main advantage of this package is a wide range of search methods implemented, including sub-optimal sequential search methods (Sequential Forward Search, SFS; Sequential Backward Search, SBS; Sequential Floating Forward Search, SFFS; Plus-L-Minus-R Search), generalized methods (Adaptive-SFFS, several Oscillating Search versions, etc.), optimal search methods like

Exhaustive Search, classical Branch and Bound (BB) and its extended versions, predictive BB search etc. This project is currently discontinued but the authors are working on a new open source C++ library (private communication).

Many specialized software tools also rely on feature selection, for example **RankGene** [11] designed to analyze gene expression data. For many more projects that include feature selection modules see the packages listed at the KDnuggets site www.kdnuggets.com.

3 Infosel++ for End-User

While Infosel++ enables easy development of new feature selection algorithms, most users are interested only in testing and comparing different already implemented algorithms. Most algorithms in this library are based on estimation of probability distributions, also several statistical algorithms (t-score, Correlation Coefficient, F-score) have been implemented for easy comparison. Different search method (ranking, forward, backward and exhaustive search) may be combined with various cost functions whenever it is appropriate.

3.1 Filter algorithms

Structure of generalized filter, wrapper and hybrid algorithms, as proposed in [12], has been implemented. For filters (Fig. 1), given a data set \mathcal{D} the search starts from a subset S (an empty set, a full set, or any randomly selected subset), exploring the space of combinations of features using particular search strategy. Each generated subset S is evaluated by some measure of relevancy $M(S)$, and if it is an improvement replaces the best subset found so far. The search iterates until a predefined stopping criterion based on information or correlation measures is reached. Changing the search strategies and evaluation measures used in steps 5 and 6 of the algorithm, new algorithms are created. The filter model applies evaluation criteria (distance, information, consistency or dependency measures) independent of predictive algorithms, therefore it avoids their biases and is computationally efficient in comparison with wrapper or hybrid algorithms.

3.2 Implemented Algorithms

Feature selection algorithms implemented in the Infosel++ have been split into 4 distinct groups (many formulas are given in [13, 6] and [14]). Acronyms in parenthesis are used in the Infosel++ menu system:

1. Ranking methods, including: CC (pcc), Pearson's Correlation Coefficient, t -score (tsc), t -score statistics (for two class problem) [15], F-score (fsc), F-score statistics (for multi-class problem) [16], χ^2 (chq), χ^2 -score statistics, MI (mi), Mutual Information, SUC (suc), Symmetrical Uncertainty Coefficient, distance rankings according to MDr (mdr) Maturity, Dr (Dr) Kilogram, KL Dr (Klara) Fullback, Bdrm (bdrm) Bhat-tacharatyya, and SDr (sdr) Sammon index.

2. Ranking with shifting of redundant features: MIFS (mifs): Mutual Information Feature Selection [17], MIFS-U (mifsu): MIFS under Uniform Information Distribution [18], AMIFS (amifs): Adaptive MIFS [19], MID (mid): Mutual Information Difference

Filter Algorithm

input: $\mathcal{D}(F_0, F_1, \dots, F_{N-1})$; a training data set with N features
 S ; a subset from which to Start the search (Starting subset)
 δ ; a stopping criterion

output: F_{best} ; Final subset selected

```
01 begin
02     initialize:  $F_{best} = S$ ;
03      $\gamma_{best} = eval(S, \mathcal{D}, M)$ ; evaluate  $S$  by an independent measure  $M$ 
04     do begin
05          $S = generate(\mathcal{D})$ ; generate a subset for evaluation
06          $\gamma = eval(S, \mathcal{D}, M)$ ; evaluate the current subset  $S$  using  $M$ 
07         if ( $\gamma \geq \gamma_{best}$ )
08              $\gamma_{best} = \gamma$ ;
09              $S_{best} = S$ ;
10     if ( $\delta$  is true) end; check stopping criterion
11     return  $F_{best} = S$ ;
12 end;
```

Fig. 1. Generalized filter algorithm.

and MIQ (miq) Quotient [16], FCD (fcd): F-test Correlation Difference and FCQ (fcq) Quotient [16].

3. Ranking with removal of redundant features: FCBF (fcfb): Fast Correlation Based Filter [20], K-S CBF (ks_cbf): A Kolmogorov-Smirnov Correlation-Based Filter [21, 22], K-SC CBF (ksc_cbf): A Kolmogorov-Smirnov Class Correlation-Based Filter [23], PRBF (prbf): Pearson Redundancy Based Filter [24].

4. Other methods include Markov Blanket approximation (mbr) [25, 26], and GD-distance ranking (gdd) [27].

These indices are based on dependency measures, information, distance and consistency measures. Ranking filters are the least expensive algorithms for feature ordering, but they cannot discover important interactions between features nor reject redundant features. Ranking with shifting of redundant features uses a heuristic based on a minimal-redundancy maximal-relevancy (MRMR) approach [17, 16] to *shift* redundant features towards less important positions. Proposed heuristics and their improvements are presented in Tab. 1. Mutual Information $MI(f_i, C)$, is one of the most common measures of dependency, $F(f_i, C)$ stands for F-score statistics and $c(f_i, f_j)$ is a correlation coefficient. β is an arbitrary parameter in range $[0, 1]$.

The second group of ranking methods selects optimal non-redundant subsets of features removing all redundant features. Liu [20] has used “predominant features” for that purpose, similar approach has been proposed by Biesiada *et al.* [21, 22], where two features are recognized as redundant if they have the same probability distributions or the

Type	Acronym	Full Name	Formula
Discrete	MIFS	Mutual info. feat. sel. [17]	$MI(f_i, C) - \beta \sum_{j \in S} MI(f_i, f_j)$
	MIFS-U	MIFS uniform distr. [18]	$MI(f_i, C) - \beta \sum_{j \in S} \frac{MI(f_i, C)}{H(f_j)} MI(f_i, f_j)$
	AMIFS	Adaptive MIFS [19]	$MI(f_i, C) - \frac{1}{\ S\ } \sum_{j \in S} \frac{MI(f_i, f_j)}{H(f_j)}$
	MID	Mutual info. difference [16]	$MI(f_i, C) - \frac{1}{\ S\ } \sum_{j \in S} MI(f_i, f_j)$
	MIQ	Mutual info. quotient [16]	$MI(f_i, C) / \frac{1}{\ S\ } \sum_{j \in S} MI(f_i, f_j)$
Continuous	FCD	F-test corr. difference [16]	$F(f_i, C) - \frac{1}{\ S\ } \sum_{j \in S} c(f_i, f_j) $
	FCQ	F-test corr. quotient [16]	$F(f_i, C) / \frac{1}{\ S\ } \sum_{j \in S} c(f_i, f_j) $

Table 1. Formulas used in Maximum Relevancy Minimum Redundancy (MRMR) ranking.

same joint distributions (feature and class) [23]. Various forward, backward and greedy search methods that use MI or SUC as evaluation functions have been implemented.

Modular construction of Infotel++ facilitates development of new methods with little coding. The technical details useful for developers will be published separately in a longer paper.

4 Illustrative results on synthetic data

Results on 3 synthetic datasets are presented here to test our implementation. They are easy to understand and point to deficiencies of some methods. Extensive tests including novel combinations of ranking and selection methods will be published elsewhere. A very simple dataset used by Shridhar *et al.* [28] contains 12 patterns with 4 features (Tab. 2). All variables in this problem are discrete, with feature $f_3 = f_2^2$ and f_4 as irrelevant, and the dependent variable y is given by $y = f_1 * f_2$. In our experiments the original dataset was copied 10 times to avoid limitations of statistical tests.

f_1	f_2	f_3	f_4	y	f_1	f_2	f_3	f_4	y	f_1	f_2	f_3	f_4	y	f_1	f_2	f_3	f_4	y
0	1	1	2	0	1	2	4	2	2	2	1	1	1	2	0	2	4	2	0
1	1	1	1	1	2	2	4	2	4	0	1	1	1	0	1	2	4	1	2
2	1	1	2	2	0	2	4	1	0	1	1	1	2	1	2	2	4	1	4

Table 2. Dataset used in analysis [28].

Synthetic ‘‘Corral dataset’’ proposed by John *et al.* [29] has been used to test relevancy and irrelevancy. It has 6 features, and the target concept is defined as the combination of $[(A0 \wedge A1) \vee (B0 \wedge B1)]$. Two additional features are Irrelevant (Orr) and Correlated (Cor), introducing 25% error rate (noise). The last of the synthetic datasets, Gauss8, has been used in our previous study [21, 24]. Gauss4 is based on sampling of 4 Gaussian functions with unit dispersion in 4 dimensions, each cluster representing a separate class. The first function is centered at (0, 0, 0, 0), the next at (1, 1/2, 1/3, 1/4), (2, 1, 2/3, 1/2), and (3, 3/2, 3, 3/4), respectively. The dataset contains 4000 vectors, 1000 per each class. Gauss8 is an extension of Gauss4, with 4 additional features that are approximately linearly dependent $f_{i+4} = 2f_i + \epsilon$, where ϵ is a uniform noise.

The summary of results for all datasets and selection algorithms is presented in Tab. 3. For the Shridhar dataset almost all ranking and redundancy shifting methods worked correctly. If the relevancy threshold is set to 0.05, features marked in bold (irrelevant)

		Shridhar dataset [28]				Corral dataset [29]				Gauss8 dataset [21, 24]									
Acronym		Most – Less Important				Most – Less Important				Most – Less Important									
Rankings	pec	f_1	f_2	f_3	f_4	cor	A_0	A_1	B_0	B_1	irr	f_1	f_5	f_6	f_2	f_7	f_3	f_4	f_8
	fsc	f_1	f_2	f_3	f_4	cor	A_0	A_1	B_0	B_1	irr	f_1	f_5	f_2	f_6	f_3	f_7	f_4	f_8
	chq	f_1	f_2	f_3	f_4	cor	A_0	A_1	B_0	B_1	irr	f_1	f_5	f_2	f_6	f_3	f_7	f_4	f_8
	mi	f_1	f_2	f_3	f_4	cor	A_0	A_1	B_0	B_1	irr	f_5	f_1	f_2	f_6	f_7	f_3	f_4	f_8
	suc	f_1	f_2	f_3	f_4	cor	A_0	A_1	B_0	B_1	irr	f_1	f_5	f_2	f_6	f_3	f_7	f_4	f_8
	mdr	f_1	f_2	f_3	f_4	cor	A_0	A_1	B_0	B_1	irr	f_1	f_5	f_2	f_6	f_3	f_7	f_4	f_8
	kdr	f_1	f_2	f_3	f_4	cor	A_0	A_1	B_0	B_1	irr	f_1	f_5	f_2	f_6	f_3	f_7	f_4	f_8
	kldr	f_1	f_2	f_3	f_4	cor	A_0	A_1	B_0	B_1	irr	f_1	f_5	f_2	f_6	f_3	f_7	f_4	f_8
	bdr	f_1	f_2	f_3	f_4	cor	A_0	A_1	B_0	B_1	irr	f_1	f_5	f_2	f_6	f_3	f_7	f_4	f_8
	sdr	f_1	f_2	f_3	f_4	cor	A_0	A_1	B_0	B_1	irr	f_1	f_5	f_2	f_6	f_3	f_7	f_4	f_8
	mifs ($\beta = 0.5$)	f_1	f_2	f_4	f_3	cor	B_0	B_1	A_0	A_1	irr	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
	mifsu ($\beta = 0.5$)	f_1	f_2	f_3	f_4	cor	B_0	B_1	A_0	A_1	irr	f_1	f_5	f_2	f_3	f_4	f_6	f_7	f_8
Redundancy	amifs	f_1	f_2	f_4	f_3	cor	B_0	B_1	A_0	A_1	irr	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
	mid	f_1	f_2	f_4	f_3	cor	B_0	B_1	A_0	A_1	irr	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
	miq	f_1	f_2	f_3	f_4	cor	A_0	A_1	B_0	B_1	irr	f_1	f_5	f_2	f_3	f_4	f_6	f_7	f_8
	fid	f_1	f_2	f_3	f_4	cor	B_0	B_1	A_0	A_1	irr	f_1	f_5	f_2	f_6	f_3	f_7	f_8	f_4
	fiq	f_1	f_2	f_3	f_4	cor	B_0	B_1	A_0	A_1	irr	f_1	f_5	f_2	f_6	f_3	f_7	f_8	f_4
	fcfb	f_1	f_2			cor	A_0	A_1	B_0	B_1	irr	f_1	f_2	f_3					
	ks_cbf	f_1	f_2			cor						f_1	f_2	f_3	f_4				
	ksc_cbf	f_1	f_2			cor						f_1	f_2	f_3	f_4				
	prbf	f_1	f_2			cor	B_0	A_1	A_0	irr		f_1	f_2	f_3	f_4				
	mbr	f_1	f_3	f_4	f_2	cor	B_1	B_0	A_1	A_0	irr	f_1	f_2	f_3	f_4	f_8	f_7	f_6	f_5
	gdd				N.A.	cor	A_1	A_2	B_1	B_0	irr	f_1	f_5	f_2	f_6	f_3	f_7	f_8	f_4

Table 3. Ordering of features after feature selection for 3 synthetic datasets.

will be automatically removed. Two algorithms "ks_cbf" and "ksc_cbf" should not really be used for this data because the Kolmogorov-Smirnov based tests are designed for continuous features only (yet they still produced reasonable results, selecting features f_1 and f_3). The same applies to the algorithm based on Markov Blanket (mbr), where two most important features selected are f_1 and f_3 . The last method listed (gdd) is very sensitive to the presence of redundant features, but it can still be used as a test for occurrences of duplicated attributes.

For the Corral dataset the optimal solution consists of only 4 features used to define the target function. All algorithms failed to identify them correctly. The correlated feature (cor) was always selected as the most important, followed by 4 relevant features, with the irrelevant feature (irr) as the least important, removed by the relevancy threshold. Algorithms based on statistical tests are again not applicable in this case.

For the Gauss8 dataset an ideal ranking method should give the following order of features: $f_1 > f_5 > f_2 > f_6 > f_3 > f_7 > f_4 > f_8$. Moreover the selection algorithms should also reject all 4 linearly dependent features as redundant leaving $f_1 > f_2 > f_3 > f_4$ order. K-S CBF and a few other methods based on statistical tests completed this task without difficulties, FCBF [20] selected 3 features only, while MI filter placed f_5 at the first position and reversed f_3 and f_7 order. All ranking methods worked as expected.

On top of the selection/ranking results for Gauss8 data described above, 4 different classification methods have been tested: Naive Bayes Classifier (NBC) (Weka implementation, [8]), the nearest neighbor algorithm (1NN) with Euclidean distance function, C4.5 tree (Weka) and the Support Vector Machine with a linear kernel (Ghostminer 3.0 implementation). The results obtained with selected subset of features are presented in Tab. 4.

Title	Selected features			
	Full set	FCBF	Ranking	K-S CBF (K-SC CBF,PRBF)
Features	1 to 8	1 to 3	1 to 8	1 to 4
NBC	82.1	81.6	82.1	82.1
1NN	73.4	68.1	73.4	73.4
C4.5	78.3	76.2	78.7	78.7
SVM	81.9	77.0	81.7	81.7
Average	79.9	75.3	80.4	80.4

Table 4. Accuracy of 4 classifiers on selected subsets of features for the Gauss8 dataset.

5 Summary

A C++ library (Infosel++) for implementing feature selection algorithms has been presented. The library provides users with the ability to easily test-drive standard feature selection methods, combining them with various predictive methods, and to develop and test novel methods. The software includes algorithms based on estimation of probability distributions, correlation coefficients, t -score and F -score, and a few additional functions for estimation of relevancy, plus a dozen of feature selection algorithms that are not easy to find in other software packages. The library, available from authors of this paper, is meant for educational, research and data mining purposes.

Due to the lack of space only results on synthetic data have been presented to verify correctness of algorithm implementation. The methods included in the library have also been tested on real datasets, such as the microarray experiments (gene expression data) and SNP data from Genome-Wide Association Studies (GWAS). The results will be presented in extended version of this paper. More details about the structure of Infosel++ library and examples of programming utilizing its functions will be presented in an extended version of this paper.

Acknowledgement: This work has been supported by the Polish Committee for Scientific Research grant 2007-2010 No.: N N519 1506 33. JB is grateful to Larry Haitkamp for fruitful discussion and support during writing of this paper.

References

1. Guyon, I., Gunn, S., Nikravesh, M., Zadeh, L.: Feature Extraction, Foundations and Applications, Studies in Fuzziness and Soft Computing Series. Springer (2006)
2. Duch, W., Maszczyk, T.: Universal learning machines. LNCS, 206-215 **5864** (2009)
3. Saeys, Y., Inza, I., Larrañaga, P.: A review of feature selection techniques in bioinformatics. Bioinformatics **23**(19) (2007) 2507–2517
4. Liu, H., (Eds), M.M.: Computational Methods of Feature Selection. CRC Press (2007)
5. Saeys, Y., Liu, H., Inza, I., Wehenkel, L., de Peer, Y.V.: New challenges for feature selection in data mining and knowledge discovery. JMLR Workshop and Conf. Proc (2008)
6. Duch, W.: Filter methods. In Guyon, I., Gunn, S., Nikravesh, M., Zadeh, L., eds.: Feature extraction, foundations and applications. Springer (2006) 89–118
7. Kohavi, R., Sommerfield, D., Dougherty, J.: Data mining using MLC++, a machine learning library in C++. Int. J. of Artificial Intelligence Tools **6**(4) (1997) 537–566

8. Witten, I., Frank, E.: *Data Mining: Practical machine learning tools and techniques*. 2nd Ed, Morgan Kaufmann, San Francisco (2005)
9. Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., Euler, T.: YALE: Rapid prototyping for complex data mining tasks. In: *Proc. of the 12th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD-06)*. (2006)
10. Pudil, P., Novovicova, J., Somol, P.: Feature selection toolbox software package. *Pattern Recognition Letters* **23**(4) (2002) 487–492
11. Su, Y., Murali, T., Pavlovic, V., Schaffer, M., Kasif, S.: Rankgene: identification of diagnostic genes based on expression data. *Bioinformatics* **19** (2003) 1578–1579
12. Liu, H., Yu, L.: Toward integrating feature selection algorithms for classification and clustering. *IEEE Trans. on Knowledge and Data Engineering* **17**(4) (2005) 491–502
13. Press, W., Teukolsky, S., Vetterling, W., Flannery, G.: *Numerical recipes in C, The art of scientific computing*. Cambridge, University Press, Cambridge, UK (1988)
14. Vilmansen, T.: Feature evaluation with measures of probabilistic dependence. *IEEE Transaction on Computers* **22**(4) (1973) 381–388
15. Golub, T., et al.: Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science* **286** (1999) 531–537
16. Ding, C., Peng, F.: Minimum redundancy feature selection from microarray gene expression data. *Journal of Bioinformatics and Computational Biology* **3**(2) (June 2004) 185–205
17. Battiti, R.: Using mutual information for selecting features in supervised neural net learning. *IEEE Trans. Neural Networks* **5**(4) (July 1994)
18. Kwak, N., Choi, C.H.: Input feature selection for classification problems. *IEEE Transactions on Evolutionary Computation* **13**(1) (2002) 143–159
19. Tesmer, M., Este'vez, P.: AMIFS: Adaptive feature selection by using mutual information. In: *Proc. of Int. Joint Conf. on Neural Networks, Budapest, IEEE Press (2004)* 1415–1420
20. Yu, L., Liu, H.: Efficient feature selection via analysis of relevance and redundancy. *Journal of Machine Learning Research, JMLR* **5** (Oct 2004) 1205–1224
21. Duch, W., Biesiada, J.: Feature Selection for High-Dimensional Data: A Kolmogorov-Smirnov Correlation-Based Filter Solution. In: *Advances in Soft Computing*. Springer (2005) 95–104
22. Biesiada, J., Duch, W.: A Kolmogorov-Smirnov correlation-based filter solution for microarray gene expressions data. In: *14th Int. Conference on Neural Information Processing (ICONIP07), LNCS. Volume 4985*. Springer (2008) 285–294
23. Blachnik, M., Duch, W., Kachel, A., Biesiada, J.: Feature Selection for Supervised Classification: A Kolmogorov-Smirnov Class Correlation-Based Filter. In: *AIMeth, Symposium On Methods Of Artificial Intelligence*. Gliwice, Poland (10-19 November 2009)
24. Biesiada, J., Duch, W.: Feature Selection for High-Dimensional Data: A Pearson Redundancy Based Filter. In: *Advances in Soft Computing*. Volume 45. Springer (2008) 242–249
25. Koller, D., Sahami, M.: Toward optimal feature selection. In: *Proc. of the 13th Int. Conf. on Machine Learning, San Francisco, Morgan Kaufmann (1996)* 284–292
26. Xing, E., Jordan, M., Karp, R.: Feature selection for high-dimensional genomic microarray data. In: *Proc. of the 8th Int. Conf. on Machine Learning*. (2001)
27. Lorenzo, J., Hernandez, M., Mendez, J.: GD: A Measure based on Information Theory for Attribute Selection. *LNCS, 124-135* **1484** (1998)
28. Sridhar, D., Barlett, E., Seagrave, R.: Informatic theoretic subset selection for neural networks models. *Computers & Chemical Engineering* **22**(4) (1998) 613–626
29. John, G., Kohavi, R., Pfleger, K.: Irrelevant features and the subset selection problem. In: *Proc. Eleventh Inter. Conf. on Machine Learning, Morgan Kaufmann (1994)* 121–129