

Feature Ranking Derived from Data Mining Process

Aleš Pilný and Pavel Kordík and Miroslav Šnorek

Department of Computer Science and Engineering, FEE,
Czech Technical University, Prague, Czech Republic
pilnya1@fel.cvut.cz, kordikp@fel.cvut.cz, snorek@fel.cvut.cz

Abstract. Most common feature ranking methods are based on the statistical approach. This paper compares several statistical methods with a new method for feature ranking derived from the data mining process. This method ranks features depending on the percentage of child units that survived the selection process. A child unit is a processing element transforming the parent input features to the output. After training, units are interconnected in the feedforward hybrid neural network called GAME. The selection process is realized by means of a niching genetic algorithm, where units connected to the least significant features starve and fade from the population. Parameters of the new feature ranking algorithm are investigated and comparison among different methods is presented on well-known real world and artificial data sets.

1 Introduction

Nowadays data with few input features is the exception. Each feature adds one dimension to the dimensionality of data vectors. For effective, more accurate data mining, it is necessary to use preprocessing methods which reduce the dimensionality of input data or describe the relevance of each feature of data. Set of methods to reduce data dimension, so called Feature Selection (FS) [12], search for a subset of relevant features from an initial set of features while Feature Extraction (FE) methods [14] create a subset of new features containing information extracted from the original set of features. Relaxed settings for FS are methods known as Feature Ranking [5], ranking of all original features in correspondence to their relevance.

Feature Selection algorithms may be divided into three categories. Algorithms in the first category are based on filters [2], where the significance of features is computed outside from the classification algorithm. On the other side Wrapper methods [6], from the second category, depend on a classifier to evaluate the quality of selected features. Finally Embedded methods [3] select relevant features within the learning process of internal parameters (e.g. weights between layers of neural networks). The goal of feature selection is to avoid selecting too many or too few variables than necessary. In practical applications, it is impossible to obtain a complete set of relevant features. Therefore, the modelled system is an open system, and all important features that are not included in the data set (for what reason ever) are summarised as noise [11].

On the other hand, it is not recommended to select as much features as possible. In fact, even if theoretically more variables should provide one with better modelling accuracy, in real cases it has been observed many times that this is not the case. This depends on the limited availability of data in real problems: successful models seem to be in good balance of model complexity and available information. In facts, variables selection tends to produce models that are simpler, clearer, computationally less expensive and, moreover, providing often better prediction accuracy [12].

In statistical analysis, forward and backward stepwise multiple regression (SMR) are widely used [12]. The resulting subset of features generated by adding features until the addition of a new feature no longer results in a significant increment in an R^2 (correlation coefficient) value.

Siedlecki and Sklansky [13] use genetic algorithms for variable selection by encoding the initial set of n variables to a chromosome, where 1 and 0 represents presence and absence respectively of variables in the final subset. They used classification accuracy, as the fitness function and obtained good neural network results.

Mutual information (MI) [3] between features can be computed by integrating the probability density functions of input and output variables. MI is very often used in FS algorithms to distinguish between useful and irrelevant features. Several FS algorithms for the WEKA [15] data mining environment are based on measuring the mutual information of attributes.

In this paper, we compare the performance of FS algorithms available in WEKA on synthetic data set generated by Tesmer and Estevez to measure the performance of the AMIFS method [14]. Then we introduce new methods for feature ranking which are byproducts of the GAME data mining algorithm [7]. We adjust their performance to improve results on syntectic data sets. Finally, we applied all FS algorithms to real-world data set.

2 Feature ranking methods in WEKA

Seven Weka FR methods will be used for comparison of ranking performance. ChiSquared method evaluates the worth of an attribute by computing the value of the chi-squared statistic with respect to the class, GainRatio by measuring the gain ratio with respect to the class, InfoGain by measuring the information gain with respect to the class, OneR by using the OneR classifier. ReliefF evaluates the worth of an attribute by repeatedly sampling an instance and considering the value of the given attribute for the nearest instance of the same and different class. SVM evaluates the worth of an attribute by using an SVM classifier and SymmetricalUncert(SU) evaluates the worth of an attribute by measuring the symmetrical uncertainty with respect to the class.

3 FeRaNGA: novel method for feature ranking

In this section, we propose a method for feature ranking derived from information gained during the data mining process. The GAME data mining engine needs to be briefly described.

3.1 Group Method of Data Handling

There are several algorithms for inductive models construction commonly known as Group Method of Data Handling (GMDH) introduced by Ivachknenko in 1966 [8]. The Multilayered Iterative Algorithm (MIA) uses a data set to construct a model of a complex system. Layers of units transfer input variables to the output of the network. The coefficients of units transfer functions are estimated using the data set describing the modeled system. Networks are constructed layer by layer during the learning stage. the Group of Adaptive Models Evolution (GAME) algorithm proceeds from the MIA algorithm. Modifications of the original algorithm are the following: maximal number of unit inputs equals to the number of layer the unit belongs to, interlayer connections are allowed, transfer function and learning algorithm of units can be of several types, niching genetic algorithm is used to select surviving units and an ensemble of models is generated. The more detailed description can be found in [7].

The niching genetic algorithm used in GAME is a cornerstone of feature ranking algorithm and needs to be described.

3.2 Niching genetic algorithm

Niching methods [10] extend genetic algorithms to domains that require the location of multiple solutions. They promote the formation and maintenance of stable subpopulations in genetic algorithms (GAs). One of these methods is deterministic crowding [9]. The basic idea of deterministic crowding is that offspring is often most similar to parents. We replace the parent who is most similar to the offspring with higher fitness.

The reason why we employ deterministic crowding instead of using just simple GA is the ability to maintain multiple subpopulations (niches) in the population. When the model is being constructed units connected to the most important input would soon dominate in the population of the first layer if we have used traditional GA(see Fig.1). All other units connected to least important inputs would show worse performance on the validation set and disappear from the population with exponential speed.

In inductive modeling we need also to extract and use information from least important features and therefore we prefer maintaining various niches in the population. The distance of genes is based on the phenotypic difference of units (to which inputs are connected). Each niche is thus formed by units connected to similar set of inputs. In the first layer just one input is allowed, therefore niches are formed by units connected to the same feature. After several epochs of GA with deterministic crowding the best individual (unit) from each niche is

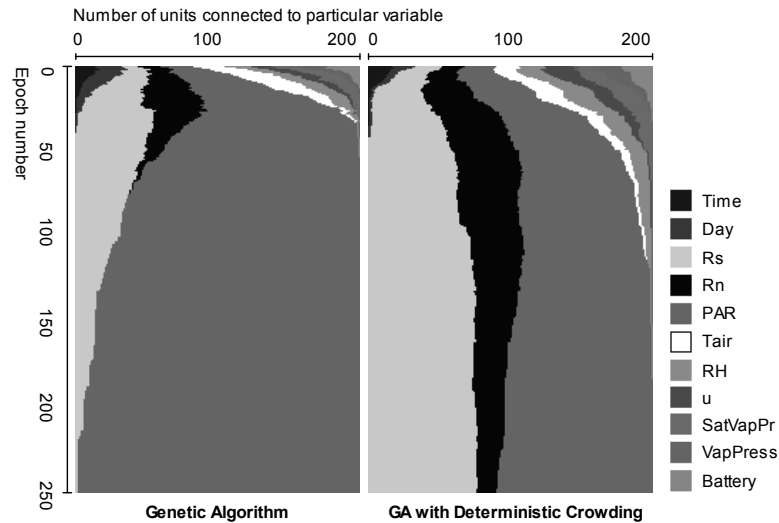


Table 1. GA versus niching GA with DC: first layer of the GAME network (units with single input)

selected to survive in the layer of the model. The construction of the model goes on with the next layers, where niching is also important.

Proposed method of feature ranking takes into account two factors. First is the significance of feature for modelling the output variable. The second is the amount of additional information to the information carried by already selected variables. This resembles to state of the art methods based on mutual information analysis. These methods select set of features of the highest mutual information with the output variable while minimizing mutual information among the selected features.

We found out that by monitoring which genes exist in the population we can estimate the significance of each feature.

3.3 Significance estimation

In the initial population of the first layer units are randomly generated. Connection to certain feature is represented as "1" in corresponding gene locus. Numbers of ones in locus are therefore uniformly distributed at the beginning of GA. After several epochs of GA with DC, numbers of ones in gene loci representing more important features increases whereas numbers in loci of least significant features decreases (see Fig. 1.).

This fact can be used for the estimation of features significance. In each layer of the network, after the last epoch of GA with DC, before the best gene from each niche is selected, we count how many genes (units) are connected to each input variable. This number is accumulated for each feature and when divided by

sum of accumulated numbers for all features, we get the proportional significance of each feature.

3.4 FeRaNGA algorithm for feature ranking

The ranking of features can be easily extracted from their proportional significance estimated by the above described procedure. This is what we call Feature Ranking utilizing information from Niching Genetic Algorithm (FeRaNGA) algorithm.

The configuration of the GAME engine, particularly size of the population and number of epochs of Niching Genetic Algorithm (NGA) has considerable influence on results of the FeRaNGA algorithm. For very low number of epochs, significance of features is close to random number, because the niching genetic algorithm is unable to eliminate units connected to irrelevant features from the population.

The GAME engine typically constructs ensemble of models [4]. We found out, that by applying FeRaNGA to all ensemble models and computing the median from estimated significance of features greatly improves the results. We will refer to this method as FeRaNGA-n where n is the number of ensemble models.

4 Data sets overview

Two artificial and one real data sets were used in experiments. Artificial data sets were generated to measure the performance of the AMFIS feature selection algorithm [14].

4.1 Gaussian Multivariate Data Set

This artificial data set consists of two clusters of points generated from two different 10th-dimensional normal Gaussian distributions. Class 1 corresponds to points generated from $N(0, 1)$ for each dimension and Class 2 to points generated from $N(4, 1)$. This data set consists of 50 features and 500 samples per class. By construction, features 1-10 are equally relevant, features 11-20 are completely irrelevant and features 21-50 are highly redundant with the first ten features. Ideally, the order of selection should be: at first relevant features 1-10, then the redundant features 21-50, and finally the irrelevant features 11-20.

4.2 Uniform Hypercube Data Set

Second artificial data set consists of two clusters of points generated from two different 10th-dimensional hypercube $[0, 1]^{10}$, with uniform distribution. The relevant feature vector $(f_1, f_2, \dots, f_{10})$ was generated from this hypercube in decreasing order of relevance from feature 1 to 10. A parameter $\alpha = 0.5$ was defined for the relevance of the first feature and a factor $\alpha = 0.8$ for decreasing the relevance of each feature. A pattern belongs to Class 1 if $(f_i < \gamma^{i-1} * \alpha / i$

= 1, . . . , 10), and to Class 2 otherwise. This data set consists of 50 features and 500 samples per class. By construction, features 1-10 are relevant, features 11-20 are completely irrelevant, and features 21-50 are highly redundant with first 10 features. Ideally, the order of selection should be: at first relevant features 1-10 (starting with feature 1 until feature 10 in the last position), then the redundant features 21-50, and finally the irrelevant features 11-20.

4.3 Housing Data Set

This real-world multivariate data set consist of 506 instances, 13 continuous attributes [1]. The last attribute, MEDV, was for experiments in Weka discretized into 4 classes, because the nominal character of the output variable is expected.

5 Experimental Analysis

The first experiment compares the performance of feature ranking methods in their implicit configuration on three different data sets. For the FeRaNGA method, we used standard configuration of the GAME engine with 15 individuals in the NGA and 30 epochs. We run the algorithm several times to show the unstable behavior.

Second part of analysis describe experiments with making the FeRaNGA algorithm more restrictive to prevent irrelevant features being selected by a chance. Finally the results of the FeRaNGA-n algorithm with well configured GAME engine are presented.

5.1 FeRaNGA algorithm vs WEKA FR methods in default settings

Most of FR methods in Weka are giving exact results corresponding to artificial data sets characteristic, except SVM method (as is shown in Table 2 and 3). In these tables only the interesting columns are displayed. The reference ranking of features is displayed in first row of tables. A gray background cells with black coloured numbers represents features which have zero significance (no units connected to them were able to survive the evolutionary process). Results of the FeRaNGA algorithm are presented under label GAME followed by number of model used for significance extraction.

For the Gaussian data set, the order in which first ten features were selected is not important, because all 10 features are equally significant. WEKA's methods (except the SVM) ranked features correctly (see Table 2). The FeRaNGA algorithm demonstrated worse results in comparison with WEKA's methods. Due to randomness of niching genetic algorithm used and insufficient number of epochs, ranks are different for each GAME model.

The Table 3 shows more or less similar results. All methods from WEKA ranked first ten features correctly, except ReliefF and SVM methods. Results of the FeRaNGA method are unstable, except the first position. The most significant feature was identified correctly for all 5 models.

Method	Relevant(1-10)										Redundant(21-50) and Irrelevant(11-20)																			
ChiSq	7	6	9	10	8	2	1	3	5	4	38	26	21	48	50	33	29	40	49	27	22	42	32	34	30	17	16	20	15	
GainRatio	10	6	5	9	7	1	3	2	4	8	38	49	27	47	29	36	33	22	37	48	24	39	43	28	34	18	20	19	17	
InfoGain	7	6	8	9	2	10	1	3	5	4	38	26	29	48	50	40	21	33	46	27	24	22	31	32	43	17	16	20	15	
OneR	7	9	10	8	2	3	1	6	4	5	38	26	25	40	29	28	50	21	24	33	35	43	30	34	27	16	11	12	14	
ReliefF	6	9	7	4	10	3	5	2	1	8	26	36	27	50	21	46	30	49	28	31	23	45	41	47	48	18	15	16	13	
SVM	2	4	6	3	7	9	1	10	8	23	47	39	30	37	42	22	34	21	38	44	5	17	41	14	28	20	45	18	25	
SU	10	6	7	9	5	1	3	2	4	8	38	47	49	27	29	33	36	48	37	22	24	44	39	40	30	18	20	19	17	
GAME 1	7	8	2	6	26	23	9	35	28	4	1	33	38	40	43	20	42	47	3	5	15	21	22	24	34	36	37	39	48	
GAME 2	10	7	3	9	37	27	24	50	8	5	23	46	44	1	2	4	6	11	12	13	19	28	29	30	36	38	39	40	47	
GAME 3	10	2	5	8	7	6	3	29	26	4	37	47	36	46	16	34	43	1	9	11	18	23	24	25	32	33	35	38	48	
GAME 4	3	9	10	41	6	7	5	8	27	25	31	48	16	39	20	34	44	28	35	1	14	21	22	23	32	33	36	37	46	
GAME 5	9	6	3	10	28	50	4	38	29	8	13	36	48	27	20	15	18	23	42	1	14	22	24	25	34	37	39	40	46	

Table 2. Ranks of features for Gaussian Multivariate Data Set used in WEKA and GAME with default settings.

Method	1	2	3	4	5	6	7	8	9	10	Redundant(21-50) and Irrelevant(11-20)																					
ChiSquare	1	2	3	4	5	6	7	8	9	10	22	37	48	35	39	43	47	26	45	27	30	49	29	32	28	33	46	31	42	17	18	14
GainRatio	1	2	3	4	5	6	7	8	9	10	23	38	27	35	37	30	34	39	28	25	22	26	50	32	36	21	31	46	33	17	18	14
InfoGain	1	2	3	4	5	6	7	8	9	10	22	37	48	43	35	39	26	47	44	30	27	49	36	29	33	28	23	46	42	17	18	14
OneR	1	2	3	4	5	6	7	8	9	10	48	37	43	50	40	30	25	44	39	45	22	49	38	34	36	33	23	28	24	18	17	14
ReliefF	1	2	3	4	8	7	9	10	5	6	29	48	47	25	31	30	49	26	27	28	21	50	24	43	45	42	23	20	44	15	41	17
SVM	1	2	48	3	29	43	4	9	35	7	36	45	41	44	34	25	8	11	12	30	37	19	15	16	6	38	24	20	5	13	40	28
SU	1	2	3	4	5	6	7	8	9	10	23	38	35	37	27	48	39	34	26	25	50	45	49	29	32	21	31	46	24	17	18	14
GAME 1	1	2	4	3	5	27	49	45	6	47	32	7	8	33	35	38	24	43	13	16	21	23	37	14	28	40	9	12	18	20	29	50
GAME 2	1	2	3	4	44	38	5	48	25	7	22	40	32	19	47	50	20	46	31	35	43	18	37	8	9	10	12	14	16	21	30	49
GAME 3	1	2	3	5	7	47	37	4	48	6	8	26	30	43	45	13	14	18	42	9	10	11	12	17	19	20	23	25	29	32	38	50
GAME 4	1	2	3	33	4	30	41	7	44	29	46	47	8	5	9	25	34	36	48	10	35	40	6	13	14	16	18	20	22	24	31	50
GAME 5	1	6	3	2	27	8	50	4	5	48	7	22	25	28	39	47	9	16	29	30	31	11	13	40	46	49	10	15	20	26	35	45

Table 3. Uniform Hypercube Data Set analysed with default settings of WEKA and GAME methods.

Method	Features Significance												Method	Features Significance											
ChiSquare	1	6	4	7	5	2	8	10	11	3	9	12	GAME 1	1	6	7	4	5	10	9	2	3	11	8	12
GainRatio	1	10	3	2	7	12	5	11	9	8	4	6	GAME 2	1	7	6	4	2	8	9	10	11	3	12	5
InfoGain	1	4	6	7	2	5	8	10	11	3	9	12	GAME 3	1	7	4	6	3	9	8	2	10	5	12	11
OneR	1	7	6	5	4	2	11	8	3	10	9	12	GAME 4	1	6	7	5	2	12	10	9	4	8	11	3
ReliefF	1	7	6	4	11	5	9	3	12	10	8	2	GAME 5	1	7	6	8	5	9	10	2	4	3	11	12
SVM	1	6	2	7	4	12	3	8	5	11	9	10	-	-	-	-	-	-	-	-	-	-	-	-	
SU	1	4	2	6	7	5	8	10	3	11	9	12	-	-	-	-	-	-	-	-	-	-	-	-	
Average	1	7	6	4	2	5	3	10	8	11	12	9	Average	1	7	6	4	2	9	5	10	8	3	12	11

Table 4. Bosthouse data GAME vs Weka

In the Table 4 we can see results for real-world data, Housing Data Set. This time, even methods from WEKA differ in their ranking. It indicates that this problem is more complex than synthetic data. All methods found first feature (Criminality in the area) as the most significant for the value of the value of housing in Boston. When the average ranking is computed from WEKA methods and from several GAME models, results are very similar.

5.2 Restricted FeRaNGA algorithm - more selective ranking

The results of the FeRaNGA algorithm applied to synthetic data (Table 2) show that some redundant features received higher ranking than expected. This behavior is caused by insufficient selection pressure in the niching genetic algorithm - number of epochs was too low to eliminate all redundant features.

In this section, we experiment with the restriction of the FeRaNGA algorithm. The ranking is computed just from best chromosomes (feature lists) from the population. Number of unique chromosomes (UNC) used for the ranking varies from 1 to all (Table 5). Results were measured on Hypercube Data Set with NGA number of epochs 150 and size of initial population 150. It can be observed, that when UNC is lower, number of used features is also reduced. For very low values of UNC only relevant features are ranked, while for all UNC redundant and irrelevant features are ranked as well.

UNC	1	2	3	4	5	6	7	8	9	10	Redundant(21–50) and Irrelevant(11–20)																		
All	1	2	3	4	5	6	46	15	24	37	48	13	17	21	28	29	30	41	43	45	49	50	7	8	9				
1/2	1	2	3	4	26	12	35	46	5	6	7	8	9	10	11	13	14	15	16	17	18	19	20	21	22				
1/3	1	2	37	21	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	22	23	24				
1/4	1	2	41	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24				
3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25				
2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25				
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25				

Table 5. Restricted FeRaNGA algorithm on the Hypercube data set. The bigger number of UNC causes the bigger number of used features.

The FeRaNGA algorithm restricted to one UNC (the best chromosome from the population) can be used to find and rank just few most important features of the data set

In the next section we present results of the FeRaNGA-n algorithm which is powered by ensemble methods.

5.3 Results for FeRaNGA-n algorithm

The GAME algorithm usually generate more models for one purpose (ensemble of models). The idea of FeRaNGA-n algorithm is to improve results of ranking

by combining unstable FeRaNGA ranks from n GAME models. Final ranks of features are computed as median ranking of features from n models.

In the Table 6 we show results of the FeRaNGA-5 algorithm on Hypercube data set. Restrictive trend corresponding with number of UNC is again evident. NGA configuration was 150 epochs and size of initial population was 150 individuals.

UNC	1	2	3	4	5	6	7	8	9	10	Redundant (21 – 50) and Irrelevant (11 – 20)													
2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1 / 4	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1 / 3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1 / 2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
2 / 3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
All	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24

Table 6. Results of FeRaNGA-n algorithm on Hypercube data set. All selected features are relevant and have correct ranks.

Results of the FeRaNGA-n are very accurate. All selected features are relevant and have correct ranks. For full number of UNC only seven from ten relevant features is selected (last row in table 6), but their ranks are accordant with their real ranks. To change the number of selected features, we can reconfigure the NGA.

5.4 Parameters of NGA for FR

The performance of the FeRaNGA-n algorithm on Hypercube data can be improved by reconfiguration of NGA parameters. The table 7 presents results of FeRaNGA-7 algorithm with all UNC and NGA's configurations displayed in the first column. First is size of initial population and second, fixed to value 30, is number of epochs.

Config.	1	2	3	4	5	6	7	8	9	10	Redundant(21-50) and Irrelevant(11-20)														
15-30	1	2	3	4	6	7	8	5	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
30-30	1	2	3	4	5	6	7	8	10	39	48	9	11	12	13	14	15	16	17	18	19	20	21	22	23
50-30	1	2	4	3	5	7	6	8	9	22	37	10	11	12	13	14	15	16	17	18	19	20	21	23	24
100-30	1	2	3	4	5	7	6	8	9	10	22	26	34	35	37	38	40	44	45	47	48	11	12	13	14
150-30	1	2	4	3	5	6	8	7	22	9	34	35	26	39	43	44	10	30	40	45	47	48	11	12	13
300-30	1	2	3	4	5	7	6	8	9	43	37	22	39	48	44	26	27	38	50	34	35	36	10	11	12

Table 7. Different configuration of FeRaNGA-7 on Hypercube data set.

When the ranks for more (or for all) features are needed, one can easily reconfigure parameters of NGA. Results in table 7 are not quite accurate, but it

can be improved by increasing number of epochs or number of GAME models from which medians are chosen.

6 Conclusion

FeRaNGA-n algorithm for feature ranking was presented and fine tuned. The comparison of FeRaNGA-n algorithm and FR methods from WEKA showed that on Hypercube data set results are equivalent, but the number of selected features depends on the configuration of NGA and on number of GAME models from which the median ranks are chosen. FeRaNGA-n algorithm can be used for feature ranking as well as for feature selection. The advantage of the algorithm is that it is primarily designed for data with continuous output variable, but it can be also used for categorical variables. It does not require any additional computation, all information for ranking are extracted from process of data mining (GAME) models evolution.

7 Acknowledgement

This research is partially supported by the grant Automated Knowledge Extraction (KJB201210701) of the Grant Agency of the Academy of Science of the Czech Republic and the research program "Transdisciplinary Research in the Area of Biomedical Engineering II" (MSM6840770012) sponsored by the Ministry of Education, Youth and Sports of the Czech Republic.

References

1. Uci machine learning repository. available at <http://www.ics.uci.edu/mllearn/MLSummary.html>, September 2006.
2. T. G. Almuallim H., Dietterich. Learning with many irrelevant features. 1991.
3. J. Biesiada, W. Duch, A. Kachel, K. Maczka, and S. Palucha. Feature ranking methods based on information entropy with parzen windows. pages 109–119, 2005.
4. G. Brown. *Diversity in Neural Network Ensembles*. PhD thesis, The University of Birmingham, School of Computer Science, Birmingham B15 2TT, United Kingdom, January 2004.
5. I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.
6. R. Kohavi. *Wrappers for Performance Enhancement and Oblivious Decision Graphs*. PhD thesis, Stanford University, 1995.
7. P. Kordík. *Fully Automated Knowledge Extraction using Group of Adaptive Models Evolution*. PhD thesis, Czech Technical University in Prague, FEE, Dep. of Comp. Sci. and Computers, FEE, CTU Prague, Czech Republic, September 2006.
8. H. Madala and A. Ivakhnenko. *Inductive Learning Algorithm for Complex System Modelling*. CRC Press, 1994. Boca Raton.
9. S. W. Mahfoud. A comparison of parallel and sequential niching methods. In *Sixth International Conference on Genetic Algorithms*, pages 136–143, 1995.

10. S. W. Mahfoud. Niching methods for genetic algorithms. Technical Report 95001, Illinois Genetic Algorithms Laboratory (IlliGaL), University of Illinois at Urbana-Champaign, May 1995.
11. J. A. Muller and F. Lemke. *Self-Organising Data Mining*. Berlin, 2000. ISBN 3-89811-861-4.
12. S. Piramuthu. Evaluating feature selection methods for learning in data mining applications. *European Journal of Operational Research*, 156:483–494, 2004.
13. W. Siedlecki and J. Sklansky. On automatic feature selection. *International Journal of Pattern Recognition*, 2:197–220, 1988.
14. M. Tesmer and P. Estevez. Amifs: adaptive feature selection by using mutual information. In *Proceedings of the 2004 IEEE International Joint Conference on Neural Networks*, volume 1, page 308, Dept. of Electr. Eng., Chile Univ., Santiago, Chile, July 2004.
15. I. Witten and E. Frank. *Data Mining – Practical Machine Learning Tools and Techniques, Second Edition*. Elsevier, 2005.