

Correlation-based Feature Ranking in Combination with Embedded Feature Selection

Aleš Pilný¹, Wolfgang Örtel², Pavel Kordík¹, Miroslav Šnorek¹

¹Dept. of Computer Science and Engineering, Karlovo nám. 13, 121 35 Praha 2, Czech Republic

²HTW-Dresden, Friedrich-List-Platz 1, D-01069 Dresden, Germany

pilnyal@felk.cvut.cz, kordik@felk.cvut.cz, snorek@felk.cvut.cz,
oertel@htw-dresden.de

Abstract. *Most of Feature Ranking and Feature Selection approaches can be used for categorial data only. Some of them rely on statistical measures of the data, some are tailored to a specific data mining algorithm (wrapper approach). In this paper we present new methods for feature ranking and selection obtained as a combination of the above mentioned approaches. The data mining algorithm (GAME) is designed for numerical data, but it can be applied to categorial data as well. It incorporates feature selection mechanisms and new methods, proposed in this paper, derive feature ranking from final data mining model. The rank of each feature selected by model is computed by processing correlations of outputs between neighboring model's neurons in different ways. We used four different methods based on fuzzy logic, certainty factors and simple calculus. The performance of these four feature ranking methods was tested on artificial data sets, on well known Ionosphere data set and on well known Housing data set with continuous variables. The results indicated that the method based on simple calculus approach was significantly worse than other three methods. These methods produce ranking consistent with recently published studies.*

Keywords

Feature Ranking, Feature Selection, Correlation, FAKE-GAME, Embedded Model.

1 Introduction

The success of data mining heavily depends on quality of input features. For some problems, input features do not contain enough information to be able to perform desired task (e.g., build accurate model or classifier). There often several possible input features that can be collected, however most of them can turn out useless. It is always better to collect more input features than to miss some crucial one. When many features are available and data records are few, course of dimensionality prevents data mining methods of working well.

Statistical methods based on mutual information analysis [1] are able to identify most relevant input features. Algorithms (e.g. AMIFS [12]) utilizing these methods can select a representative subset of informative non-correlated features helping to overcome the curse of dimensionality.

The main drawback of these methods is that they are primary designed for nominal (discrete) variables and classification problems. In this paper, we propose computational intelligence methods for feature selection and ranking, that are applicable to numerical features and regression problems as well.

At first, we would like to clarify the difference among the feature ranking, feature selection and feature extraction. The feature ranking process only ranks all features in correspondence to their relevance while feature selection methods create a subset of the most relevant features. This subset should provide a maximal amount of information from the original subset without any redundant or irrelevant features. Methods of feature extraction, create a subset of new features by extracting the information from the original set of features.

While feature ranking simply assign a rank (relevance) to each feature regardless of their interrelations, feature selection solves a different problem - to choose the best subset of features. Note that in this subset should not contain redundant features.

Generally, it is possible classify feature selection algorithms into filters, wrappers and embedded approaches [2]. Filters evaluate quality of selected features independently from the classification algorithm, while wrapper methods depend on a classifier to evaluate quality of selected features. Finally embedded methods [2] selects relevant features within a learning process of internal parameters (e.g. weights between layers of neural networks).

The goal of this paper is to describe new methods for feature ranking where these methods ranks only features pre-selected by the embedded feature selection algorithm. This embedded approach is based on special type of an artificial neural network, the GAME neural network [6].

Each method, we are proposing, ranks features using different approach, but all of them are based on inter-correlations inside the network. Feature ranking and selection process is always performed independently.

2 Embedded feature selection process

Embedded feature selection process is an integral part of proposed feature ranking methods and needs to be briefly described. This process is implemented in the FAKE-GAME [6] tool for data mining and knowledge discovery.

2.1 GAME network

A base of the FAKE-GAME tool is the Group of Adaptive Models Evolution algorithm (GAME) producing GAME networks (data mining models). The algorithm is a modification of the Multilayered Iterative Algorithm (MIA). The MIA belongs to algorithms for inductive models construction, commonly known as Group Method of Data Handling (GMDH) [8] and 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. Main differences between MIA and GAME are 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, an ensemble of models is generated and finally the most important improvement - a genetic algorithm is used to optimize the topology. The more detailed description about the FAKE-GAME can be found in [6].

2.2 Feature selection process

Before feature ranking, the most significant features are selected. The GAME network is constructed by using a niching genetic algorithm - the corner stone of this selection 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. The parent is replaced by an offspring with higher fitness, and the most similar genotypic information. The reason why authors 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 one have used traditional GA. 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 one need also to extract and use information from least important features and therefore maintaining various niches in the population is preferred. 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 and 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 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.

Finally we obtain the subset of features which are useful for solving the given problem. The fact that a feature is used (selected) means that it contains important information for output determination. Therefore only significant features are selected as inputs to the network and than one may compute the importance of each feature. Redundant and irrelevant features are eliminated in the genetic algorithm.

The GAME algorithm is also used in feature ranking method FeRaNGA [11] where ranks of selected features are derived from proportional numbers of connected individuals in genetic algorithms optimizing layers of units. Generally, the importance of feature increases by an amount of additional information to the information carried by already selected variables.

3 Correlation based feature ranking methods

In previous section we have described the way how to create a subset of important features. When we need to know an importance of selected features as well, then we can analyze the topology of generated GAME network. The topology consists of different types of units (neurons with different transfer functions). When the network is ready, we know all outputs of all inner units (responses of neurons to input data vectors presented to the network). Rank of each feature is in our approach obtained as a relationship between this feature and the whole network output. As a measure of a relationship determination we used a correlation coefficient.

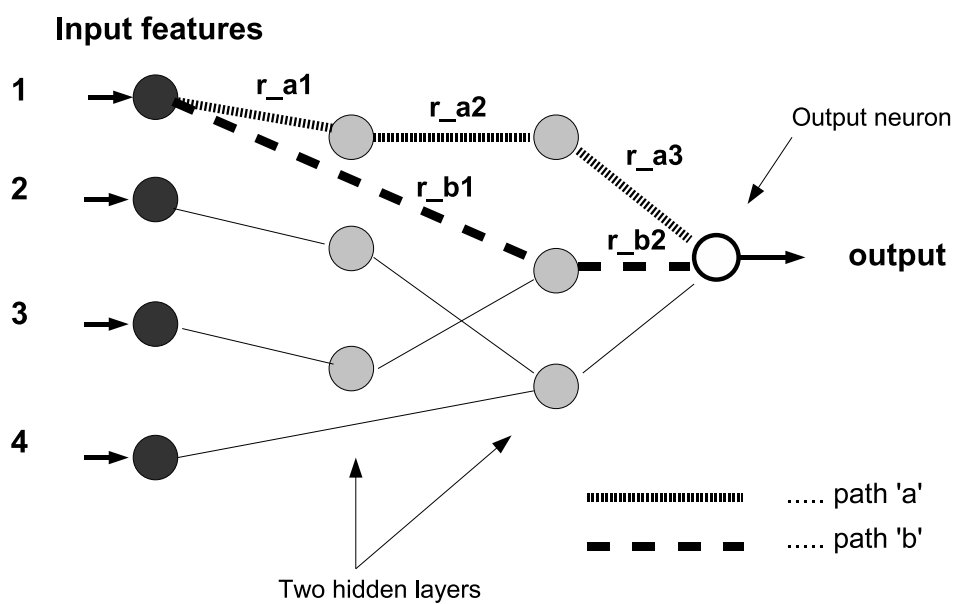


Fig. 1. Example of the GAME network structure with four input features, one output neuron and two hidden layers. For example, path 'a' and path 'b' (highlighted) have different length. r_{a1}, r_{a2}, r_{a3} are correlation coefficients (intercorrelations) between neighbouring neuron outputs among the path 'a' and r_{b1}, r_{b2} have the same meaning among the path 'b'.

In probability theory and statistics, correlation indicates the strength and direction of a linear relationship between two random variables [5]. This relationship is often measured as a correlation coefficient. The correlation coefficient is 1 in

the case of an increasing linear relationship, -1 in the case of a decreasing linear relationship, and some value in between in all other cases, indicating the degree of linear dependence between the variables. The closer the coefficient is to either -1 or 1, the stronger the correlation between the variables. If the variables are independent then the correlation is 0.

The proposed methods use all neighbour neuron output intercorrelations among the path between input feature and output of the network. From the definition of the GAME network is possible to have more than one path between input and output of the network (see example in Fig. 1). These methods differ in a way of intercorrelations processing and are described in following subsections. Note that we are using correlation coefficients in absolute values because we need the merit of relationship only, not the sign. These methods based on next listed approaches: simple mathematical operations, fuzzy logic and certainty factors.

3.1 Simple mathematical approach - MP-FR

To derive an influence of individual input feature to the output, we analyze the structure of the network. Input features are not connected directly to the output thereby processing of internal relationships between neighbouring units - intercorrelations - is needed. One simplest way is multiplying of neighbouring correlations along each of the path between input feature and output. Finally add up all these multiples to get final number - significance of currently processed feature. Simply Multiply-Plus - Feature Ranking (MP-FR). This method has disadvantage because of different path length possibility and various number of paths among the output and separate features. This disadvantage might be removed by normalizing. Significance S_1 of input feature nr.1 from example in Fig.1 by using MP-FR is computed as:

$$S_1 = (r_{a1} * r_{a2} * r_{a3}) + (r_{b1} * r_{b2})$$

General formula for computing of feature significance is:

$$S_i = \sum_{l=1}^N \prod_{j=1}^K r_{lj}$$

where r_{lj} is j -th inter-correlation between neighbour neuron outputs on path number l .

3.2 Fuzzy logic approach - FL-FR

Possible disadvantages of MP-FR method can be easily removed by sophistic method where correlations along the path are not multiplied. We are finding the best relationship between input feature and output. A correlation represents here a measure of neighbouring relation. The most important relation along this path is a minimal relationship, the minimal correlation. More than one path means also more of minima. Therefore is necessary to find the maximum of all minima among the paths between input feature and output. This process is very similar to operations from the fuzzy set theory, specially to standard complement and standard union, introduced by L. Zadeh in 1965 [13]. Therefore this approach is called Fuzzy Logic - Feature Ranking (FL-FR). Computation of significancy S_i for feature i can be formalised as:

$$S_i = \max(\min(r_{11}, \dots, r_{1K_1}), \dots, \min(r_{N1}, \dots, r_{NK_N}))$$

where r_{NK_N} is intercorrelation between neighbour neurons on path nr. N and K_N is K -th inter-correlation on the same path.

3.3 Certainty Factor approach

In the 1980s, Dvid McAllister, developed a metric for 'certainty factors' for use in an 'expert system' (a type of computer program)[4]. A certainty factor is used to express how accurate, truthful, or reliable one judge a predicate to be. It is one's judgement of how good the evidence is. The issue is how to combine various judgements. Let's consider a hypothesis, H, and evidence, E. The rule for evaluation is:

$$IF E \text{ is observed THEN } H \text{ is true (with cetainty factor, } CF = n)$$

In McAllister's scheme, a certainty factor is a number (n in the rule above) from 0.0 to 1.0 (it reflects evidence for the hypothesis only). A phrase such as 'suggestive evidence' is given a number such as 0.6; 'strongly suggestive evidence' is

given a number such as 0.8. The person making the judgement uses the scale more or less as an ordinal scale. The numbers were used in a metric to permit a computer to make calculations. McAllister's rules for combining certainty factors are such that one can add new evidence to existing evidence. If the evidence is positive, this increases that certainty, as one would expect. But one never become 100 percentual certain.

In our case the certainty factor is an absolute value of inter-correlation between neighbouring neuron outputs. There are two approaches how to use the certainty factors for computing feature importance. One way is use basic certainty factor judgement (chaining certainty factors) and other way is to use combining certainty factors.

3.3.1 Basic Certainty Factors approach - BCF-FR

When CF's occur in sequence, the resulting CF is found by multiplying the CF's in the chain. Since the premises together have CF = 0.5, and the rule CF is 0.8, then the conclusion is just the product of the two values. This procedure we have used for determining the conclusion for each separate path between the current input feature and output of the network. Final importance of processed feature, S_i , can be then computed as a maximum of conclusions for separate paths - similar process as in rules with ORed premises - (formalised in following equation).

$$S_i = \max\left(\prod_{j=1}^{K_1} r_{1j}, \dots, \prod_{j=1}^{K_N} r_{Nj}\right)$$

where r_{Nj} is j-th inter-correlation between the neighbouring neuron outputs on path number N .

3.3.2 Combine Certainty Factors approach - CCF-FR

In this method are certainty factors combined along the paths and rank is assigned in dependency on maximal value of conclusion. The equation for adding two positive neighbouring certainty factors (j-th and (j+1)-th) on path N is:

$$CF_{cobmi}(r_{Nj}, r_{Nj+1}) = r_{Nj} + (1 - r_{Nj}) * r_{Nj+1}$$

and importance of input feature i is then maximum of all conclusions on paths between feature i and output of the network:

$$S_i = \max(CF_{combi1}, \dots, CF_{combiN})$$

where CF_{combiN} is result on N -th path.

4 Experimental data sets

We have performed various experiments on different data sets. Two artificial data sets and one real word dataset were used.

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 and was created by M. Tesmer and P. A. Estevez for experiments in [12]. 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 $(f_i < \gamma^{i-1} * \alpha / i = 1, \dots, 10)$, 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. This data set also come from [12].

4.3 Ionosphere real-world data set

This radar data (from ML UCI repository [3]) was collected by a system in Goose Bay, Labrador. This system consists of a phased array of 16 high-frequency antennas with a total transmitted power on the order of 6.4 kilowatts. The targets were free electrons in the ionosphere. "Good" radar returns are those showing evidence of some type of structure in the ionosphere. "Bad" returns are those that do not; their signals pass through the ionosphere. Received signals were processed using an autocorrelation function whose arguments are the time of a pulse and the pulse number. There were 17 pulse numbers for the Goose Bay system. Instances in this database are described by 2 attributes per pulse number, corresponding to the complex values returned by the function resulting from the complex electromagnetic signal. Number of Instances is 351, number of attributes 34 and one class attribute. All predictor attributes are continuous.

4.4 Housing real-world data set

This Boston Housing Dataset (from ML UCI repository [3]) was taken from the StatLib library which is maintained at Carnegie Mellon University. Number of instances is 506 and number of attributes is 13. Attributes are continuous.

5 Experiments

Four various experiments were performed. Two on artificial data sets and two on real-world data sets. All experiments have the same first step - generating of five data mining models over the data where subsets of the most significant features are selected. These subsets differ among the models because of random initialization of niching genetic algorithm (used for model cration). Configuration of this genetic algorithm was identical for all experiments.

In the first two experiments we have tested ranking ability of proposed methods on artificial data sets. Results on Gaussian Multivariate data Set (classification problem) shows table 1. Each of five generated models selects just two features and all of them were equally relevant (according to the data definition) except redundant feature 29 in model nr. 3. Selection of feature Nr 29 might be casused by random initialization. But each method assigned ranks correctly for these features (firstly relevant feature Nr 7 then redundant feature Nr 29).

Second experiment, done on Uniform Hypercube data set (classification problem) in the same way as first experiment, obviously showed the power of proposed approach. The selection process took into account only the most important feature (only feature Nr 1 was selected) and showed us how important the selection step is. Results are not shown for its simplicity.

Third experiment was focused on comparison of proposed methods on Ionosphere real-world data set. As in two previous experimets five models were generated for obtaining of selected features subsets and then all new methods for feature ranking were applied to. Number of features in subsets can differ because of random initialization of niching genetic algorithm. Gained ranking results for each method were then processed separately and finally compered. The base measure for comparing ranking results is classification accuracy (CA). From each ranked subset (each method results for each model) were generated new ten models from which CA was then computed as a mean value of individual CA's. Moreover, CA's were computed in the same way also for subsequently decreasing number of features from original subsets. The number decreases from maxima in odd numbers to minimal subset size two.

Table 2 shows win and lost ratio for Ionosphere real-world data set used in this experiment. This ratio was computed on statitically significant results of t-student test from CA mean difference among all methods. The differences were

Tab. 1. Ranking ability results on Gaussian Multivariate data set for all proposed methods. All selected features (each model selects only two features) in each model are relevant except model Nr 3 where feature Nr 29 is redundant. Assigned ranks of selected features are correct in all cases.

| Method \ Model Nr | 1 | 2 | 3 | 4 | 5 |
|-------------------|-----|------|------|-----|-----|
| MP-FR | 7 8 | 3 10 | 7 29 | 2 4 | 6 1 |
| FL-FR | 7 8 | 3 10 | 7 29 | 2 4 | 1 6 |
| BCF-FR | 7 8 | 3 10 | 7 29 | 2 4 | 6 1 |
| CCF-FR | 7 8 | 3 10 | 7 29 | 2 4 | 1 6 |

Tab. 2. Win and lost counts for all methods and all subsets on Ionosphere real-word data set. MP-FR and BCF-FR have significantly worse results than FL-FR and CCF-FR. The best results proved CCF-FR method.

| No. of feat. | | 13 | 11 | 9 | 7 | 5 | 3 | 2 | (win - lost) |
|--------------|------|----|----|---|---|---|----|---|--------------|
| MP-FR | win | 1 | 2 | 1 | 2 | 2 | 1 | 1 | -19 |
| | lost | 0 | 2 | 6 | 6 | 4 | 7 | 4 | |
| FL-FR | win | 0 | 1 | 3 | 4 | 2 | 6 | 5 | 10 |
| | lost | 2 | 2 | 0 | 0 | 4 | 2 | 1 | |
| BCF-FR | win | 0 | 1 | 2 | 2 | 2 | 0 | 4 | -10 |
| | lost | 0 | 2 | 2 | 3 | 2 | 8 | 4 | |
| CCF-FR | win | 1 | 2 | 2 | 2 | 4 | 11 | 1 | 15 |
| | lost | 0 | 0 | 0 | 3 | 2 | 1 | 2 | |

computed within a subsets with the same number of features. From the difference between win's and lost's (last column) is clear that the significantly most successful method is CCF-FR and second one, also with good result, method FL-FR. On the contrary methods MP-FR and BCF-FR have significantly worse score of difference between win's and lost's. On the basis of this results we have tested only FL-FR and CCF-FR method on next real data set.

Results for fourth experiment (table 3) describes the comparison on RMS error between our proposed methods (FL-FR and CCF-FR) and ICA-FX method from [7] on real-word Housing data set (note, this is regression problem). Results for ICA-FX are averages of five regression methods (MLP, SVM, 1-NN, 3-NN and 5-NN described in [7]). All methods (also our proposed methods) were tested 10 times and the numbers in the parentheses are the averages of standard deviations of the 10 experiments corresponding to each regression method. The second row of each algorithm shows the best performance among the five regression methods (for ICA-FX method) or the best performance among the ten runs of FL-FR and CCF-FR methods.

It is clear that FL-FR and CCF-FR have comparable results as ICA-FX and in some cases better (as less standard deviation or smaller RMS error for specific Nr of features).

Tab. 3. Comparison on real word Housing data set - RMS error between FL-FR, CCF-FR and ICA-FX. Results for ICA-FX are averages of five regression methods (MLP, SVM, 1-NN, 3-NN and 5-NN). Each method (also our proposed methods) was tested 10 times and the numbers in the parentheses are the averages of standard deviations of the 10 experiments corresponding to each regression method. The second row of each algorithm shows the best performance among the five regression methods (for ICA-FX) or the best performance among ten runs of specific algorithm (FL-FR and CCF-FR).

| Method \ no. of features | 2 | 3 | 5 | 7 | 8 | 9 | 11 |
|--------------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| FL-FR | 3.78 (0.08) | 3.93 (0.41) | 3.15 (0.23) | 3.9 (0.32) | 3.75 (0.2) | - | - |
| | 3.65 | 3.64 | 2.91 | 3.55 | 3.45 | - | - |
| CCF-FR | 5.79 (0.05) | 3.98 (0.08) | 4.08 (0.41) | 3.48 (0.28) | 4.51 (0.52) | - | - |
| | 5.71 | 3.9 | 3.79 | 3.2 | 3.761 | - | - |
| ICA-FX | - | 4.09 (0.53) | 3.74 (0.51) | 3.37 (0.55) | - | 3.48 (0.63) | 3.61 (0.72) |
| | - | 3.35 (MLP) | 3.43 (5-NN) | 3.25 (3-NN) | - | 3.20 (MLP) | 3.27 (SVM) |

6 Conclusion

New approaches for feature ranking algorithms in combination with feature selection process were presented. In contrast to classical feature ranking methods these methods use only a subset of features preselected by an embedded feature selection mechanism in the FAKE-GAME. Next advantage of these proposed methods is robustness for different problems as are classification and regression. Experiments showed that significantly robust are methods based on fuzzy logic (FL-FR) and based on combined certainty factors (CCF-FR). These methods also showed the ability to be successfully compared to other regression methods.

References

- [1] R. Battiti. Using mutual information for selecting features in supervised neural net learning. *IEEE TRANSACTIONS ON NEURAL NETWORKS*, 5, NO. 4, 1994.
- [2] 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.
- [3] C. J. M. C. L. Blake. Uci repository of machine learning databases. <http://www.ics.uci.edu/mllearn/MLSummary.html>, September 2006.
- [4] R. J. Chassell. About certainty factors. <http://www.rattlesnake.com/notions/certainty-factors.html>, 2009.
- [5] W. A. N. Joseph Lee Rodgers. Thirteen ways to look at the correlation coefficient. *The American Statistician*, 42, 1988.
- [6] 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.
- [7] N. Kwak, C. Kim, and H. Kim. Dimensionality reduction based on ica for regression problems. *Neurocomputing*, 71:2596–2603, 2008.
- [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] A. Pilný, P. Kordík, and M. Snorek. Feature ranking derived from data mining process. *18th International Conference on Artificial Neural Networks - ICANN 2008*, pages 889–898, 2008.
- [12] 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.
- [13] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.