Applied Artificial Intelligence, 00:1–24

ISSN: 0883-9514 print/ISSN 1087-6545 online
DOI: 10.1080/08839510903363487

SELECTION OF NUMERICAL AND NOMINAL FEATURES BASED ON PROBABILISTIC DEPENDENCE BETWEEN FEATURES

Krzysztof Michalak[1*] **and Halina Kwasnicka**[2] **and Ewa Watorek**[3] **and Marian Klinger**[3]

[1] *Wroclaw University of Economics, Institute of Business Informatics, 53-345 Wroclaw, Poland*

[2] *Wroclaw University of Technology, Institute of Informatics, 50-370 Wroclaw, Poland*

[3] *Wroclaw Medical University, Department and Clinic of Nephrology and Transplantation Medicine, 50-417 Wroclaw, Poland*

Abbreviated title: Selection of Numerical and Nominal Features

Data classification tasks often concern objects described by tens or even hundreds of features. Classification of such high-dimensional data is a hard computational problem. Feature selection techniques help reduce the amount of computations and improve classification accuracy.

In (Michalak and Kwasnicka 2006a;b) we proposed a feature selection strategy that selects features in an individual or pairwise manner based on the assessed level of dependence between features. In the case of numerical features this level of dependence can be expressed numerically using linear correlation coefficient. In this paper the feature selection problem is addressed in the case of a mixture of nominal and numerical features. The feature similarity measure used in this case is based on the probabilistic dependence between features. This similarity function is used in an iterative feature selection procedure which we proposed for selecting features prior to classification. Experiments prove that using the probabilistic dependence similarity function along with the presented feature selection procedure can improve computation speed while preserving classification accuracy in the case of mixed nominal and numerical features.

Keywords: feature similarity, similarity measures, feature selection, pairwise selection, pattern classification

*Corresponding author. Email: krzysztof_michalak@poczta.onet.pl

1. Introduction

Classification of objects is an important task in many research areas and practical applications. The goal of classification is to assign one of several classes to each object from a given set. Objects in a classification task may be characterized by specifying values of some features for each object. Classification of multidimensional data is usually hard because some of the features may be irrelevant or misinformative and due to the phenomena often referred to as the curse of dimensionality which occur in high-dimensional spaces and have a negative impact on computation results (Verleysen 2003).

One of the approaches to dimensionality reduction is feature selection. This process should be performed in such a way that features which discriminate well between the classes are left untouched while the others are dropped. It is not an easy task because there are no generic methods of selecting features that are best suitable for classification.

In the case of objects characterized by some features the space containing all classified objects can be described formally as a product $\mathcal{X} = \prod_{f \in F} X_f$, where F is an index set which indexes the features and X_f is the domain of feature f. A classification of objects from $\mathcal{X}$ is a function $c : \mathcal{X} \to \mathcal{C}$ with values in the class set $\mathcal{C}$. In the later part of this paper we will identify the features with the elements of the index set F. Thus the feature selection problem can be formally stated as the problem of selecting a subset $F' \subset F$ of the index set. Usually, feature selection is based on some quantitative criterion $Q : 2^F \to \mathrm{R}$ that is used for measuring the capacity of a feature set $F' \subset F$ to discriminate between the classes.

Depending on how many features constitute a set F' evaluated by the criterion Q, feature selection techniques are divided into univariate and multivariate ones. Univariate methods have relatively low computational complexity, however they do not take into account any possible interdependencies between features. Multivariate feature selection methods may select more appropriate feature sets when features are correlated. However, they are usually far more computationally complex. Another disadvantage of the multivariate methods is that the feature selection process itself may suffer from the effects of high data dimensionality if the sample

size is small.

Another criterion by which feature selection methods can be divided is whether the classifier is used during the feature selection process. With respect to that aspect feature selection methods are divided into filters and wrappers (Kohavi and John 1997). Filters choose the best feature set based solely on the properties of the features. Wrappers on the other hand, measure the performance of a preselected classifier on a training data set and select features that minimize classification error on this set given by that particular classifier. Computational complexity of wrapper methods is higher than complexity of filter methods because the classifier needs to be evaluated on some data samples. The advantage of wrapper methods is that they do not use predefined selection rules to choose good features. They are thus more flexible and can work well for a wider range of data sets.

In our previous work we have introduced a Correlation Based Feature Selection (CBFS) method (Michalak and Kwasnicka 2006a;b) which is an iterative wrapper approach. The CBFS method uses linear correlation coefficient which is suitable for measuring dependence level in the case of numerical data. In this paper we propose a solution for the case of mixed nominal and numerical data. The results of the experiments show that our approach produces results not worse than the ones obtained by existing methods and is more computationally efficient.

In section 2 the iterative approach to feature selection is presented. Section 3 discusses issues concerning measuring dependency between nominal features and presents examples of non–redundant correlated features. Section 4 describes the evaluation function proposed for use in selection of nominal and numerical features. Section 5 contains the summary of experimental results, section 6 concludes the paper.

2. Selection Strategy

The number of all possible subsets of the feature set is usually so high that evaluating all possible feature subsets is not feasible. Therefore, iterative approaches are often used. The most straightforward feature selection strategies are individual ranking (Kittler 1978), forward search and backward search. Apart from the above

mentioned simple techniques of feature selection, more sophisticated methods have also been taken into consideration by some authors. For example, a genetic algorithm in which the population consists of potential feature sets and the fitness is calculated using the criterion Q has been proposed (Kwasnicka and Orski 2004, Yang and Honavar 1998). Other approaches are hybrid methods (Das 2001, Xing et al. 2001).

In all the described approaches, the criterion Q is used for evaluating the quality of the feature set $F' \subset F$. In the case of filter methods this criterion is based on the properties of features themselves. In the case of wrapper methods classification results obtained using the actual learning model are used to calculate $Q(F')$.

In this paper we use the criterion Q derived from the average classification error obtained on the training data set V_{train} using only features from the evaluated subset F'.

$$Q(F') = 1 - E(F', V_{train}) \ . \tag{1}$$

In order to find the best feature set we maximize this criterion.

All greedy approaches (individual ranking, forward search and backward search) can lead to suboptimal results because they do not evaluate all the possible feature subsets. If individual features are added to selected features subset possible dependencies between features are taken into account only to very small extent which can further deteriorate the results (Cover and Van Campenhout 1976, Pudil et al. 1994).

As a trade-off between computational complexity and the necessity to take into consideration as much dependencies between features as possible, a Pairwise Selection Strategy (PFS) was proposed (Harol et al. 2007, Pekalska et al. 2005). This selection strategy has a quadratic complexity which is reasonable for moderate numbers of features. In the pairwise selection strategy the procedure begins with an empty set $F_0 = \emptyset$, and then features are added in pairs to the set of already selected features. In each selection step n, the pair that maximizes the criterion Q

together with the already selected features is added.

The selection process is stopped when the number of features $|F_n|$ is equal to the predetermined number l. The pairwise selection strategy is thus a greedy forward search, but it allows the dependencies between features to be more thoroughly examined than in the case of individual forward searching. This advantage, however is achieved at the cost of much higher computational complexity.

The PFS strategy is aimed at taking dependencies between features into account to a higher extent than individual selection strategy does. The main disadvantage of the PFS method is that it evaluates all features in a pairwise manner which is considerably faster than evaluating all possible feature sets but can still require long computations for data sets with many features. The improvement of computational efficiency can be achieved if some of the features are evaluated individually and some in a pairwise manner.

In the CBFS method which we proposed for numerical features (Michalak and Kwasnicka 2006a;b) the decision whether to evaluate a feature individually or in pairs with other features is based on the correlation of this feature with other features. If a given feature correlates significantly with any other feature it should be evaluated in pairs with all the other features. If, on the other hand, the feature does not correlate significantly with any other feature it can be evaluated individually. The criterion Q defined by Eq. (1) is used for evaluating an individual feature or a pair of features together with the previously selected features. Obviously, if $|F_{n-1}| = l - 1$ (where l is the required number of features) only one feature has to be added to the feature set and thus no pairwise evaluation is performed. The Correlation Based Feature Selection strategy (CBFS) has lower complexity than PFS method and produces at least equally good results in terms of classification error (Michalak and Kwasnicka 2006a;b).

In the CBFS method the enumeration process is controlled by a fixed-value parameter θ which defines the threshold between individual and pairwise evaluation of features. The higher the value of θ is the more likely it is that features will be evaluated individually. Setting $\theta < 0$ causes all features to be evaluated in a pairwise manner while setting $\theta = 1$ causes all features to be evaluated individually. Therefore, in general, increasing the value of θ is expected to fasten the compu-

tations at the expense of classification accuracy, while setting lower values of θ is expected to produce better classification accuracy by performing a slower but more thorough selection process.

The complexity of selecting l features in a pairwise manner is $O(l^2)$ and the complexity of selecting l features individually is $O(l)$ thus the correlation–based feature selection strategy has lower complexity than the PFS search for a sufficiently large l and a suitably large θ.

The approach described above can be generalized by using a general evaluation function δ instead of the linear correlation coefficient. This function should quantitatively express the level of dependence between features. If for a given feature f_i there exists at least one other feature f_j such that the value of function $\delta(f_i, f_j)$ exceeds θ then the feature f_i is evaluated in a pairwise manner with all the features in the remaining set. Otherwise the feature f_i is evaluated individually.

3. Measuring Dependence of Nominal Features

Similarity function based on feature correlation seems to work well for numerical features, however it performs relatively poorly for nominal features. This is hardly surprising, considering that correlation is not, in fact, well defined for features that are not unambiguously ordered. For example, in the well known Iris data set (available from the UCI repository of machine learning databases (Blake et al. 1998)) samples are divided into three classes (flower species). The class can be treated as a nominal attribute (no ordering is predefined on these flower species). Correlation between the attribute f_2 "sepal width in cm" and the class is -0.4194 if classes are numbered 0, 1 and 2 in the same order in which they appear in the data (Fig. 1). However, if classes are numbered 2, 0 and 1 the correlation is 0.6112 (Fig. 2). From this example it is clear that dependence of nominal features should be measured in some way other than using linear correlation coefficient.

There are a number of similarity measures used for nominal features, such as mutual information and its various normalized versions. However, most of them have some undesirable properties. For example, mutual information itself and MI–induced metric $d(X,Y) = H(X,Y) - I(X;Y)$ are not bounded by a value which is

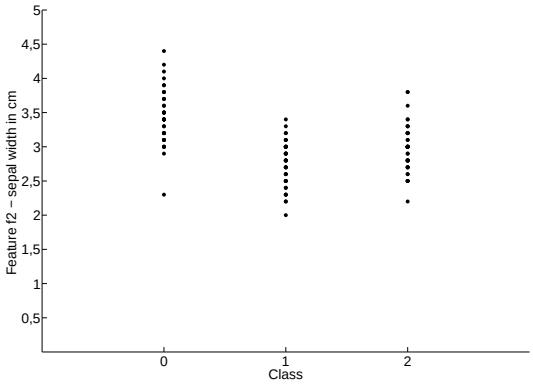

Figure 1 Feature f_2 values and the classes numbered in the original order in the Iris data set. Correlation -0.4194

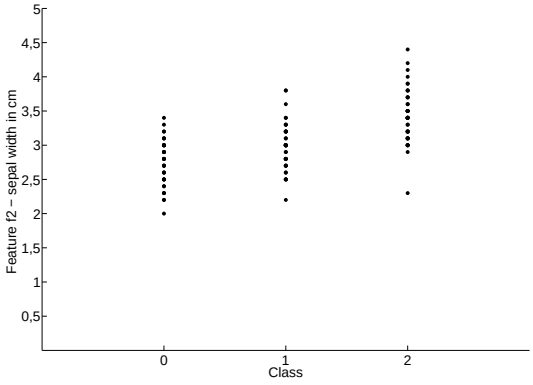

Figure 2 Relation between feature f_2 values and the classes permuted by $\binom{0\ 1\ 2}{2\ 0\ 1}$ in the Iris data set. Correlation 0.6112

constant for all features in a given data set and therefore it is problematic to select a good decision boundary for MI–based criterion. Other functions such as uncertainty coefficient are asymmetric. Feature selection methods based on redundancy, symmetric uncertainty and normalized metric $D(X,Y) = d(X,Y)/H(X,Y)$ were tested but we have not obtained results significantly better than with PFS method.

Is is worth noticing that even a very strong dependence between two features does not imply redundancy. It is quite common in literature that correlated features are treated as redundant (eg. (Gasca et al. 2006, Liu and Schumann 2005, Masaeli et al. 2010)). Some authors penalize feature subsets if there are dependencies among them (eg. (Hall 1998)). However, it is easy to construct examples of correlated features that have no discriminatory power by themselves but allow a perfect classification when considered in pairs (Fig. 3 and 4).

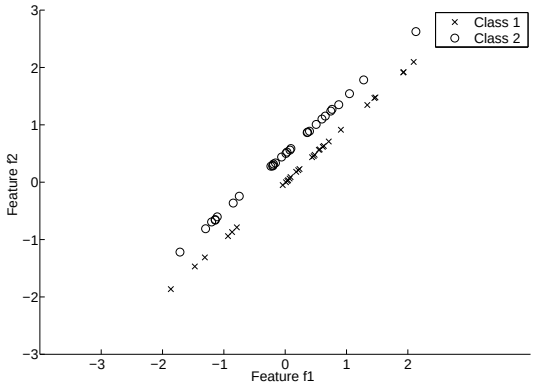

Figure 3 Example of non–redundant features with correlation 0.97

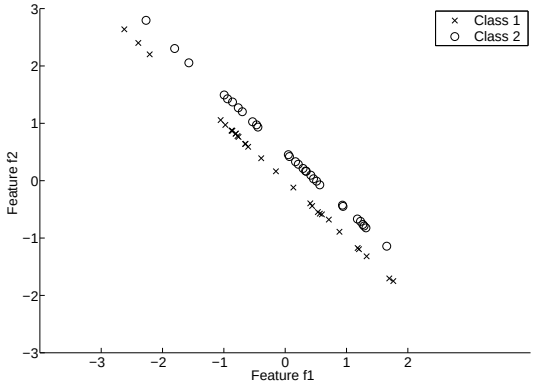

Figure 4 Example of non–redundant features with correlation −0.97

4. Similarity Function for Nominal and Numerical Features

In this paper we propose a similarity function that can be used in the iterative feature selection process described in the previous section. This function is designed to work well for nominal features. It can also be easily applied to the case when both nominal and numerical features are present by performing a discretization of continuous numerical parameters.

The proposed function is based on probabilistic dependence of features treated as random variables. In the case of correlation the similarity function $\delta(f_i, f_j)$ reaches 0 when the features are uncorrelated and 1 when features are linearly correlated. In the case of the probabilistic dependence similarity function (PDSF) we assume that the function should reach 0 when the two features treated as random variables are independent. The condition for independence of two features is:

$$P(f_i = x_n, f_j = y_m) = P(f_i = x_n) \cdot P(f_j = y_m) \ . \tag{2}$$

Therefore, we can set:

$$\delta_0(f_i, f_j) = \sum_{m,n} \Big(P(f_i = x_n, f_j = y_m) - P(f_i = x_n) \cdot P(f_j = y_m) \Big)^2 \ . \tag{3}$$

where P represents probabilities estimated on the training set.

We also assume that the case when two features f_i and f_j are "the most dependent" is when for each value of feature f_i the feature f_j takes exactly one unique value. Formally, it can be described as the situation when there exists a "1–to–1" mapping:

$$m : X_{f_i} \to X_{f_j} \tag{4}$$

such that:

$$\forall k \ \ f_j(k) = m(f_i(k))$$

where k is the number of the sample and $f_i(k)$ and $f_j(k)$ denote the values of features f_i and f_j respectively in the sample with number k. For such features we will ommit the quantifier and simply write $f_j = m(f_i)$.

For two isomorphic features f_i and f_j we would like the δ function to reach 1. To achieve this, we can divide the function $\delta_0(f_i, f_j)$ by the function $\Delta(f_i, f_j)$ which is a form of $\delta_0(f_i, f_j)$ obtained when $f_j = m(f_i)$. Thus we have:

$$\begin{aligned}\Delta(f_i, f_j) = & \\ \delta_0(f_i, f_j) = & \sum_{m,n} P(f_i = x_n, f_j = y_m)^2 \\ & - 2\sum_{m,n} P(f_i = x_n) \cdot P(f_j = y_m) \\ & \quad \cdot P(f_i = x_n, f_j = y_m) \\ & + \sum_{m,n} P(f_i = x_n)^2 \cdot P(f_j = y_m)^2 \ .\end{aligned} \tag{5}$$

Considering Eq. (4) we have:

$$P(f_i = x_n, f_j = y_m) = P(f_i = x_n) = P(f_j = y_m) \ \text{ for } \ y_m = m(x_n)$$

and

$$P(f_i = x_n, f_j = y_m) = 0 \ \text{ for } \ y_m \neq m(x_n).$$

Thus:

$$\begin{aligned}P(f_i = x_n, f_j = m(x_n)) = P(f_i = x_n) = & \\ = P(f_j = y_m) = P(f_j = m(x_n))\end{aligned} \tag{6}$$

$$P(f_i = x_n, f_j \neq m(x_n)) = 0 \tag{7}$$

Substituting Eq. (7) in Eq. (5) and considering that Eq. (4) is "1–to–1" we have:

$$\begin{aligned}\Delta(f_i, f_j) = &\sum_n P(f_i = x_n, f_j = m(x_n))^2 \\ &- 2\sum_n P(f_i = x_n) \cdot P(f_j = m(x_n)) \\ &\cdot P(f_i = x_n, f_j = m(x_n)) \\ &+ \sum_{m,n} P(f_i = x_n)^2 \cdot P(f_j = y_m)^2 \ .\end{aligned} \tag{8}$$

which is equivalent to:

$$\begin{aligned}\Delta(f_i, f_j) = &\frac{\sum_n P(f_i = x_n)^2 + \sum_m P(f_j = y_m)^2}{2} \\ &- \sum_n P(f_i = x_n)^3 - \sum_m P(f_j = y_m)^3 \\ &+ \sum_{m,n} P(f_i = x_n)^2 \cdot P(f_j = y_m)^2\end{aligned} \tag{9}$$

because from Eq. (6) it follows that all three factors under the summation sign in the second and third line of Eq. (8) can be substituted by any term from Eq. (6) to the power of 3.

Because for isomorphic features f_i and f_j we want to have $\delta(f_i, f_j) = 1$ we have to normalize the function $\delta_0(f_i, f_j)$ by the factor Eq. (9) which represents the value of the function $\delta_0(f_i, f_j)$ for isomorphic features and we get the equation for PDSF function $\delta(f_i, f_j)$:

$$\delta(f_i, f_j) = \frac{\delta_0(f_i, f_j)}{\Delta(f_i, f_j)} \ ,$$

where $\delta_0(f_i, f_j)$ is defined by equation Eq. (3) and $\Delta(f_i, f_j)$ is defined by equation Eq. (9).

From Eq. (2) and Eq. (3) it follows that when two features f_i and f_j are independent we have:

$$\delta_0(f_i, f_j) = 0$$

and thus:

$$\delta(f_i, f_j) = 0$$

and when two features f_i and f_j are isomorphic from the derivation Eq. (5)-(9) we have:

$$\delta_0(f_i, f_j) = \Delta(f_i, f_j)$$

and thus:

$$\delta(f_i, f_j) = 1$$

Therefore, the PDSF function can be used to numerically express the similarity between features and has some properties similar to those of the correlation, i.e. it reaches 0 for independent features and 1 for features that are isomorphic.

The same function can be used to measure the similarity between both nominal and numerical features. To allow this, numerical features must be discretized to a preset small number of values. Discrete values are only used to calculate the value of the function $\delta(f_i, f_j)$, the classification process is performed on original features.

5. Experiments

In the experiments the iterative feature selection procedure described in our previous work (Michalak and Kwasnicka 2006a;b) was used with the PDSF similarity

function. Classification errors and computation times were compared to those obtained using the PFS method.

The experiments were performed on four datasets. Three datasets (CRX, TicTacToe and Votes) are available at the UCI Repository of Machine Learning Databases (Blake et al. 1998). Fourth dataset FACS contains real-life medical data.

Basic characteristics of all data sets are presented in Table 1. This table also summarizes the parameters of the selection process. In TicTacToe and Votes datasets all features are nominal. The CRX data set consists of mixed nominal and numerical features.

Data set	CRX	FACS	TicTacToe	Votes
Total samples	653	123	958	435
Training samples	20,40,60	20,40,60	10,20,50	10,20,50
	80,100	80,100	100,200,500	100,200
Total number of features	15	25	9	16
Number of nominal features	9	5	9	16
Number of real–valued features	3	13	0	0
Number of int–valued features	3	7	0	0
Maximum number of selected features	14	24	8	16
Number of classes	2	3	2	2

Table 1. Data sets used in the experiments

The FACS data set contains real-life medical data gathered from patients with renal diseases. Studied population consisted of 123 patients in various stages of chronic nephropathies: 60 patients with chronic kidney disease (CKD) and 63 patients with end–stage renal disease 44 of them hemodialysed (HD), and 19 on peritoneal dialysis (PD). In the experiments classification to CKD, HD and PD groups was tested. Studied population was characterized by 25 demographic and biochemical parameters described in Table 2.

Abbreviations used in Table 2 have the following meaning:

CEC - circulating endothelial cells

CRP - C-reactive protein

EPC - endothelial progenitor cells

HB - hemoglobin

HDL - high density lipoproteins

HT - hematocrit

LDL - low density lipoproteins

Parameter	Parameter type
Gender	nominal
Age	integer
MDRD	real
Creatinine	real
Diabetes	nominal (boolean)
Hypertension	nominal (boolean)
Anti–hypertensive treatment type	nominal
Statins treatment	nominal (boolean)
EPC (control value)	integer
EPC	integer
CEC (control value)	integer
CEC	integer
VEGF	real
Endostatin	integer
WBC	real
RBC	real
HB	real
HT	real
CRP	real
Proteins	real
Serum albumins	integer
Cholesterol	real
HDL	real
LDL	real
TG	real

Table 2. Measured parameters in FACS experiments (features in the FACS data set)

MDRD - formula for calculation of renal function impairment

RBC - red blood cells

TG - triglycerides

VEGF - vascular endothelial growth factor

WBC - white blood cells

Several of measured parameters are directly related to certain groups eg. creatinine, MDRD. Some parameters such as age are unrelated to the groups. Chronic nephropathy is connected with:

- rise in serum creatinine reaching maximum levels in HD and PD,

- decrease of MDRD with minimal values in PD and HD

- gradually decreasing hemoglobin following impairment of renal function and resulting necessity for treatment with erythropoietin (EPO) to maintain proper hemoglobin level (11,0 mg/dl).

In patients with CKD, HD and PD level of circulating endothelial progenitor cells (EPC) is decreased in contrast to healthy persons and is decreasing with renal impairment (Choi et al. 2004). The levels of endostatin which is a potent

EPC inhibitor are increasing with renal function impairment (unpublished data). Some other parameters as hypertension, diabetes, hypercholesterolemia, malnutrition measured by decreased serum albumins are correlated with renal diseases, however they do not discriminate between CKD, HD and PD groups.

Measured parameters represent the whole variety of possible feature types. Some of the parameters are real–valued, some of them are discrete numerical features. There are 5 nominal features 3 of which are boolean.

In the experiments data classification was performed using the following classifiers:

Bayesian classifiers (NLC and NQC) - in the case of data sets with binary classification. NLC (Normal density based Linear Classifier) (Duda et al. 2000) is defined as:

$$f(x) = [x - \frac{1}{2}(m_1 + m_2)]^T S^{-1}(m_1 - m_2) + log\frac{p_1}{p_2} \ ,$$

where m_1 and m_2 are the estimated means for classes, S is the estimated covariation matrix and p_1 and p_2 are the a priori class probabilities.

NQC (Normal density based Quadratic Classifier) is defined as:

$$\begin{aligned} f(x) = & \tfrac{1}{2}(x - m_1)^T S_1^{-1}(x - m_1) \\ & - \tfrac{1}{2}(x - m_2)^T S_2^{-1}(x - m_2) \\ & + \tfrac{1}{2} log\tfrac{|S_1|}{|S_2|} + log\tfrac{p_1}{p_2} \end{aligned}$$

where m_1 and m_2 are the estimated means for classes, S_1 and S_2 are estimated covariation matrices for classes and p_1 and p_2 are the a priori class probabilities.

Neural network classifiers - multilayer perceptron with 5 and 20 hidden neurons and with two different numbers of output neurons N_{out} (one output neuron and the number of output neurons equal to the number of classes). These two variables generate a total of four different network architectures. Smaller network with $N_{hid} = 5$ was sized so that the number of hidden neurons was roughly equal to the

geometric mean of maximum number of input neurons and the number of output neurons. Number of hidden neurons in the larger network $N_{hid} = 20$ was set to be roughly equal to the maximum number of features (and thus input neurons) in the data sets.

Sigmoid activation function was used in output neurons and hyperbolic tangent activation function was used in hidden neurons. Network weights were optimized using SCG (Scaled Conjugate Gradient) method (Moller 1993).

Decision tree classifiers - two decision tree construction algorithms were used: ID3 and C4.5 (Quinlan 1993). C4.5 algorithm performs decision tree prunning during which some features may be left unused. Thus, C4.5 algorithm may be considered as an embedded feature selection example. Nevertheless PDSF feature selection was performed in this case in the exactly same way as with the other classifiers. Possible exclusions of features by C4.5 algorithm were done independently of PDSF choices.

The parameter θ used to choose individual or pairwise feature evaluation was set to $\theta = 0.25$. This parameter sets the balance between computation speed and classification accuracy. The value of 0.25 was found to be suitable for all data sets to speed computations up without loosing classification accuracy compared to the PFS method. In the experiments PDSF was used for all feature pairs (ie. for evaluating similarities between two nominal, two numerical and also between one nominal and one numerical feature). Continuous numerical features were discretized to 10 equally spaced bins only for calculation of the δ function. This simple approach was considered sufficient for estimating distributions of feature values, however, it is possible to employ more sophisticated methods such as adaptive binning. Undiscretized data were used for classification.

From each data set $|V_{train}|$ samples were used for classifier training and the remaining ones were used as a test set. Experiments were performed with the number of features to be selected l set to each even number $l = 2, \ldots, l_{max}$ to compare the effectiveness of the selection methods for various percentages of selected features. For each data set, each classifier, each number of features to be selected and for each number of training samples 50 iterations of the train-test cycle were performed. In

each train-test cycle a different permutation of features was used and the training set was randomly selected to avoid the influence of the initial feature ordering and the training set choice on the test results.

Tables 3-6 present the average classification errors (E_{PDSF} and $E_{PFS)}$) and average calculation durations (T_{PDSF} and $T_{PFS)}$) obtained for each data set using various classifiers and with different number of samples in the training set.

Classifier	$\|V_{train}\|$	E_{PDSF}	E_{PFS}	T_{PDSF} [s]	$T_{PFS}[s]$
C4.5	20	0.3949	**0.3945**	**8.6078**	9.8546
-/-	40	0.3660	**0.3640**	**22.9272**	25.5315
-/-	60	0.3576	**0.3544**	**37.5128**	45.6437
-/-	80	**0.3200**	0.3322	**48.5158**	63.0241
-/-	100	**0.3319**	0.3432	**59.1220**	76.3685
ID3	20	**0.3935**	0.3972	**12.3360**	15.2319
-/-	40	**0.3348**	0.3367	**18.6553**	21.0043
-/-	60	**0.3169**	0.3218	**22.1414**	27.3932
-/-	80	**0.3149**	0.3177	**26.2655**	34.1864
-/-	100	0.3119	**0.3104**	**30.5140**	40.9690
MLP ($N_{hid} = 5$)	20	**0.2818**	0.2857	**93.7648**	95.5925
-/-	40	0.2410	**0.2382**	**90.2757**	108.2888
-/-	60	**0.2021**	0.2066	**96.1499**	116.3956
-/-	80	**0.1900**	0.1917	**102.0411**	124.9733
-/-	100	0.1919	**0.1908**	**106.3514**	144.7749
MLP ($N_{hid} = 20$)	20	**0.2784**	0.2799	**102.9831**	121.2309
-/-	40	**0.2230**	0.2244	**122.1062**	137.9949
-/-	60	**0.2006**	0.2026	**128.4176**	165.3029
-/-	80	**0.1960**	0.1962	**157.1094**	189.0355
-/-	100	**0.1882**	0.1902	**153.8172**	214.1278
MLP ($N_{hid} = 5, N_{out} = 1$)	20	0.3461	**0.3437**	**83.6254**	94.2883
-/-	40	**0.2532**	0.2552	**75.0524**	94.8021
-/-	60	0.2338	**0.2330**	**94.2286**	115.6596
-/-	80	**0.2103**	0.2105	**88.5350**	117.6962
-/-	100	**0.1926**	0.1957	**85.1331**	121.4908
MLP ($N_{hid} = 20, N_{out} = 1$)	20	**0.3158**	0.3288	**95.6794**	113.9463
-/-	40	0.2788	**0.2715**	**117.0471**	132.3408
-/-	60	0.2671	**0.2654**	**146.4041**	164.1309
-/-	80	**0.2313**	0.2350	**148.8513**	169.6206
-/-	100	**0.2153**	0.2179	**153.7867**	189.9188
NLC	20	0.2181	**0.2172**	0.1821	**0.1797**
-/-	40	0.1652	**0.1651**	0.1970	**0.1908**
-/-	60	**0.1494**	**0.1494**	**0.1937**	0.2047
-/-	80	0.1448	**0.1444**	**0.1771**	0.2174
-/-	100	**0.1401**	0.1407	**0.1524**	0.2324
NQC	20	**0.2134**	0.2151	**0.1673**	0.1780
-/-	40	**0.1564**	0.1565	**0.1835**	0.1940
-/-	60	**0.1488**	0.1489	**0.1743**	0.2062
-/-	80	**0.1423**	**0.1423**	**0.1966**	0.2178
-/-	100	0.1378	**0.1377**	**0.2030**	0.2348
All classifiers		**0.2449**	0.2463	**63.2446**	77.3219

Table 3. Average classification errors and execution times for the CRX data set.

For CRX data set (Table 3) the lowest classification errors are given by Bayesian classifiers. Neural networks peform only a little worse. Decision trees give the highest classification errors. The PDSF–based feature selection method is in most cases faster than the PFS approach. It is only a bit slower for the smallest number of samples in the training data set and the fastest classifier - the NLC. This effect is caused by the overhead of calculating the δ function which for a very fast feature selection process becomes a significant factor.

The best classifiers for FACS data set (Table 4) are Bayesian classifiers. They give classification errors between 0.15 and 0.25 which is quite good for a three–class classification. The PDSF–based feature selection process is in most cases faster than PFS approach. However, because Bayesian classifiers are very fast and PDSF–based feature selection requires some additional calculations the advantage is not very big when these classifiers are used and even in three cases the PFS

Classifier	$\lvert V_{train}\rvert$	E_{PDSF}	E_{PFS}	T_{PDSF} [s]	$T_{PFS}[s]$
C4.5	20	**0.4185**	0.4192	**70.2426**	86.6476
-/-	40	0.3929	**0.3912**	**175.2097**	219.7755
-/-	60	0.3490	**0.3469**	**283.1926**	370.1299
-/-	80	**0.3148**	0.3166	**415.7696**	538.1070
-/-	100	0.3175	**0.3146**	**600.7607**	714.5004
ID3	20	**0.5886**	0.5929	**103.7889**	120.3103
-/-	40	0.4893	**0.4887**	**111.3141**	133.4603
-/-	60	0.4685	**0.4648**	**149.5528**	175.9785
-/-	80	**0.4477**	0.4577	**177.4071**	221.5509
-/-	100	**0.4461**	0.4495	**197.5555**	266.1130
MLP ($N_{hid} = 5$)	20	0.3622	**0.3600**	**394.1304**	428.2219
-/-	40	0.3171	**0.3090**	**380.0666**	474.4062
-/-	60	**0.2885**	0.2933	**439.9015**	520.2276
-/-	80	0.2767	**0.2760**	**442.4149**	558.8686
-/-	100	**0.2759**	0.2785	**506.9259**	597.9119
MLP ($N_{hid} = 20$)	20	**0.3573**	0.3639	**422.1378**	498.7840
-/-	40	**0.3118**	0.3135	**514.3097**	608.3338
-/-	60	**0.2935**	0.2975	**577.4194**	714.0221
-/-	80	**0.2805**	0.2808	**621.6012**	818.4038
-/-	100	0.2662	**0.2624**	**752.4600**	903.3907
MLP ($N_{hid} = 5, N_{out} = 1$)	20	**0.5557**	0.5564	**324.6841**	401.9815
-/-	40	0.4812	**0.4796**	**352.4628**	433.3781
-/-	60	0.4273	**0.4265**	**378.5304**	457.4927
-/-	80	0.4120	**0.4090**	**418.7992**	481.2472
-/-	100	**0.3795**	0.3916	**451.6808**	505.0900
MLP ($N_{hid} = 20, N_{out} = 1$)	20	0.5949	**0.5921**	**443.3061**	475.3185
-/-	40	**0.5096**	0.5224	**434.8527**	556.2262
-/-	60	0.4817	**0.4767**	**487.0969**	634.8327
-/-	80	**0.4550**	0.4602	**619.5457**	716.8855
-/-	100	**0.4197**	0.4286	**645.6617**	787.9557
NLC	20	**0.2311**	0.2315	**1.1224**	1.1964
-/-	40	**0.1949**	0.1984	**1.1894**	1.2920
-/-	60	**0.1605**	0.1619	1.4028	**1.3965**
-/-	80	0.1416	**0.1408**	**1.5277**	1.5446
-/-	100	**0.1482**	0.1486	1.7321	**1.7102**
NQC	20	0.2473	**0.2460**	0.9044	**0.8644**
-/-	40	**0.1972**	0.1975	**0.8416**	0.9369
-/-	60	0.1542	**0.1536**	**0.9699**	1.0010
-/-	80	**0.1512**	0.1519	**0.9197**	1.0857
-/-	100	**0.1517**	0.1536	**1.0880**	1.1773
All classifiers		**0.3439**	0.3451	**297.6120**	360.7939

Table 4. Average classification errors and execution times for the FACS data set.

approach has proven to be a little faster than PDSF–based approach

In the case of TicTacToe data set (Table 5) all classifiers perform similarly. The best classifier seems to be C4.5 but it has only a minimal advantage over all the other classifiers. The classification error for this data set is relatively high. The PDSF–based method produced a significant efficiency gain for this data set and classification error values for both examined methods are similar.

Classification of samples in the Votes data set (Table 6) seems to be quite easy - classification error values are the lowest among all datasets tested. For the Votes data set the PDSF–based method performs equally good as the PFS method in terms of classification error and shortens computation time by about 30% on average.

The experiments have shown that the proposed method PDSF is much faster than the PFS approach. To show that the PDSF–based method gives on average not worse classification results than the PFS method we performed a T–Student

Classifier	$\lvert V_{train}\rvert$	E_{PDSF}	E_{PFS}	T_{PDSF} [s]	$T_{PFS}[s]$
C4.5	20	0.4538	**0.4532**	**0.5813**	1.3658
-/-	40	**0.4079**	0.4083	**0.3013**	2.2638
-/-	60	0.3683	**0.3680**	**0.3427**	2.6928
-/-	80	0.3548	**0.3541**	**0.3735**	3.0304
-/-	100	**0.3296**	0.3340	**0.4077**	3.2425
ID3	20	**0.5153**	0.5224	**1.9555**	3.2668
-/-	40	0.5371	**0.5349**	**1.3052**	4.5052
-/-	60	0.5406	**0.5372**	**1.3817**	5.7537
-/-	80	0.5426	**0.5423**	**1.5644**	7.0112
-/-	100	**0.5433**	0.5447	**1.7678**	8.2619
MLP ($N_{hid} = 5$)	20	**0.4273**	0.4323	**12.9165**	31.6356
-/-	40	**0.3978**	0.3998	**4.6875**	34.0956
-/-	60	**0.3741**	0.3821	**3.9478**	36.6408
-/-	80	**0.3661**	0.3694	**3.8942**	39.0223
-/-	100	**0.3498**	0.3516	**4.1637**	41.3997
MLP ($N_{hid} = 20$)	20	**0.4262**	0.4329	**16.4804**	37.5259
-/-	40	**0.3902**	0.3917	**6.1917**	44.5446
-/-	60	**0.3748**	0.3753	**6.4105**	54.3732
-/-	80	**0.3539**	0.3548	**5.9395**	58.8133
-/-	100	**0.3436**	0.3463	**6.6527**	66.0443
MLP ($N_{hid} = 5, N_{out} = 1$)	20	**0.5017**	0.5086	**12.4206**	30.7438
-/-	40	**0.4560**	0.4625	**4.2605**	32.9848
-/-	60	**0.4208**	0.4249	**3.2859**	34.8172
-/-	80	**0.3948**	0.3976	**3.5795**	36.1812
-/-	100	**0.3666**	0.3702	**3.6564**	38.0626
MLP ($N_{hid} = 20, N_{out} = 1$)	20	**0.5396**	0.5406	**14.7179**	36.4227
-/-	40	**0.4768**	0.4802	**5.4201**	42.8344
-/-	60	**0.4296**	0.4334	**5.1983**	51.1083
-/-	80	0.4098	**0.4081**	**5.3944**	54.3658
-/-	100	0.3822	**0.3744**	**6.1081**	61.3974
NLC	20	**0.4230**	0.4239	**0.0426**	0.0801
-/-	40	**0.4011**	**0.4011**	**0.0123**	0.0716
-/-	60	**0.3783**	0.3793	**0.0104**	0.0717
-/-	80	0.3700	**0.3685**	**0.0100**	0.0742
-/-	100	**0.3631**	0.3638	**0.0107**	0.0783
NQC	20	**0.4191**	0.4209	**0.0322**	0.0627
-/-	40	**0.3918**	0.3947	**0.0105**	0.0668
-/-	60	0.3792	**0.3779**	**0.0098**	0.0711
-/-	80	0.3758	**0.3737**	**0.0102**	0.0738
-/-	100	**0.3691**	0.3698	**0.0107**	0.0782
All classifiers		**0.4161**	0.4177	**3.6367**	22.6284

Table 5. Average classification errors and execution times for the TicTacToe data set.

test with the null hypothesis that the average error m_{PDSF} given by the PDSF–based method is greater than the average error m_{PFS} given by the PFS method. In other words we try to reject the null hypothesis $H_0 : m_{PDSF} > m_{PFS}$ at the confidence level α with the alternative hypothesis $H_1 : m_{PDSF} \leq m_{PFS}$ based on the values of classification errors E_{PDSF} and E_{PFS} measured in the experiments.

P-value obtained from the statistical test is the upper bound of probability of mistakenly assuming that the null hypothesis is false when it is in fact true. If the value α calculated from experimental results is low it can be safely assumed that the PDSF–based method gives on average no higher classification errors than the PFS method.

In tables 7 and 8 the calculated values of α (p-value) are presented. These tables also present the average values of $E_z = E_{PFS} - E_{PDSF}$ and $T_z = T_{PFS} - T_{PDSF}$. In the case of all except one classifier the calculated p-values suggest with 90% certainty that the PDSF–based method gives not worse classification results than

Classifier	$\lvert V_{train}\rvert$	E_{PDSF}	E_{PFS}	T_{PDSF} [s]	$T_{PFS}[s]$
C4.5	20	**0.1034**	0.1038	2.8445	3.5650
-/-	40	**0.0839**	0.0850	3.7400	5.4078
-/-	60	**0.0757**	0.0775	4.8826	7.3034
-/-	80	0.0762	**0.0761**	5.6787	8.5665
-/-	100	**0.0699**	0.0710	6.1246	9.2959
ID3	20	0.3414	**0.3388**	22.8269	28.2419
-/-	40	0.2647	**0.2530**	26.2339	39.1744
-/-	60	**0.2306**	0.2359	32.3821	49.9912
-/-	80	**0.2413**	0.2449	40.6637	61.7864
-/-	100	**0.2315**	0.2352	48.1227	77.6953
MLP ($N_{hid} = 5$)	20	0.1077	**0.1076**	90.2877	112.1622
-/-	40	**0.0760**	0.0771	105.3510	147.5974
-/-	60	**0.0707**	0.0708	120.5790	175.0650
-/-	80	**0.0673**	0.0683	132.6794	193.5235
-/-	100	0.0659	**0.0653**	140.7328	211.7728
MLP ($N_{hid} = 20$)	20	**0.0923**	0.0925	111.0426	142.8561
-/-	40	**0.0804**	0.0810	142.9041	197.8154
-/-	60	**0.0750**	0.0766	173.5502	251.2506
-/-	80	**0.0677**	0.0690	210.7263	305.2989
-/-	100	0.0662	**0.0647**	250.6118	372.9950
MLP ($N_{hid} = 5, N_{out} = 1$)	20	0.1384	**0.1370**	121.0328	156.4844
-/-	40	**0.0958**	0.0964	116.2375	171.2491
-/-	60	0.0764	**0.0746**	120.0523	181.3276
-/-	80	0.0768	**0.0755**	129.6430	189.2821
-/-	100	**0.0694**	0.0700	134.8077	199.0715
MLP ($N_{hid} = 20, N_{out} = 1$)	20	**0.1850**	0.1860	145.9564	186.4691
-/-	40	0.1242	**0.1222**	151.6581	224.1439
-/-	60	0.0956	**0.0932**	179.2729	268.9543
-/-	80	**0.0821**	0.0826	199.4825	297.1225
-/-	100	0.0779	**0.0775**	221.7711	325.1803
NLC	20	0.0876	**0.0870**	0.2863	0.3140
-/-	40	**0.0689**	0.0715	0.2704	0.3384
-/-	60	0.0502	**0.0500**	0.2767	0.3620
-/-	80	**0.0456**	0.0458	0.2870	0.3871
-/-	100	0.0459	**0.0456**	0.3139	0.4146
NQC	20	**0.0868**	0.0877	0.2820	0.3145
-/-	40	**0.0599**	0.0603	0.2748	0.3393
-/-	60	0.0514	**0.0510**	0.2781	0.3644
-/-	80	**0.0446**	0.0448	0.2838	0.3857
-/-	100	0.0438	**0.0436**	0.3075	0.4133
All classifiers		**0.1024**	**0.1024**	79.8685	115.1071

Table 6. Average classification errors and execution times for the Votes data set.

Classifier	E_z	Error p-value	T_z	Time p-value
C4.5	$7.83 \cdot 10^{-4}$	0.04	32.53	≤ 0.01
ID3	$1.54 \cdot 10^{-3}$	0.09	20.08	≤ 0.01
MLP ($N_{hid} = 5$)	$5.91 \cdot 10^{-4}$	0.15	53.68	≤ 0.01
MLP ($N_{hid} = 20$)	$1.44 \cdot 10^{-3}$	≤ 0.01	83.44	≤ 0.01
MLP ($N_{hid} = 5, N_{out} = 1$)	$1.11 \cdot 10^{-3}$	0.09	50.62	≤ 0.01
MLP ($N_{hid} = 20, N_{out} = 1$)	$1.74 \cdot 10^{-3}$	0.02	72.99	≤ 0.01
NLC	$4.46 \cdot 10^{-4}$	≤ 0.01	$4.55 \cdot 10^{-2}$	≤ 0.01
NQC	$2.46 \cdot 10^{-4}$	0.07	$5.86 \cdot 10^{-2}$	≤ 0.01
All classifiers	$1.05 \cdot 10^{-3}$	≤ 0.01	42.43	≤ 0.01

Table 7. p-values for the null hypothesis calculated per classifier.

Classifier	E_z	Error p-value	T_z	Time p-value
CRX	$1.41 \cdot 10^{-3}$	≤ 0.01	14.08	≤ 0.01
FACS	$1.16 \cdot 10^{-3}$	≤ 0.01	63.18	≤ 0.01
TicTacToe	$1.60 \cdot 10^{-3}$	≤ 0.01	18.99	≤ 0.01
Votes	$5.44 \cdot 10^{-5}$	0.45	35.24	≤ 0.01
All classifiers	$1.05 \cdot 10^{-3}$	≤ 0.01	42.43	≤ 0.01

Table 8. p-values for the null hypothesis calculated per data set.

PFS method. For half of the classifiers the certainty is even higher - above 95%. For MLP classifier with $N_{hid} = 5$ the statistical certainty that the PDSF–based method is better than PFS is 85% which is considered quite low by statistical standards but such result does not indicate that the PDSF–based method is worse than PFS

either.

The results grouped by data set show that PDSF–based method performs very well on most data sets (including the real-life FACS data set) with statistical certainty over 99%. For the Votes data set the results are inconclusive. The overall results for all classifiers and all data sets are very promising. The statistical certainty of the hypothesis that the PDSF–based method gives not worse classification results than the PFS method reaches 99%.

A quick comparison was performed to compare PDSF feature selection with correlation based feature selection method (CBFS) described in our previous work (Michalak and Kwasnicka 2006a;b). In this part of experiments calculations on all four data sets were performed using two the most ubiquitous classifiers: NLC and C4.5. In these tests 10 iterations were performed for each set of parameters, each data set and each classifier. Value of $E_z = E_{CBFS} - E_{PDSF}$ was recorded, where E_{CBFS} and E_{PDSF} are average classification errors obtained using each method. Results obtained in the experiments are shown in Table 9.

Classifier	E_z for NLC	E_z for C4.5
CRX	0.000350340	0.00193260
FACS	0.000107170	0.01418500
TicTacToe	0.000930960	0.00047708
Votes	0.000021127	0.00039395
All classifiers	0.00024563	0.00587200

Table 9. Average difference $E_{CBFS} - E_{PDSF}$ between classification errors given by CBFS and PDSF methods.

Total E_z value calculated for both classifiers equals $E_z = 0.0027943$ and the p-value for the null hypothesis that the correlation–based feature selection method gives better results is less than 0.01. Thus, the experiments performed prove with high statistical significance that the new criterion designed for data sets with nominal and mixed features performs better than the previous correlation–based approach.

6. Conclusion

In this paper a new feature similarity function based on probabilistic dependence between features is proposed. The probabilistic-dependence similarity function (PDSF) can be used in an iterative feature selection process to decide whether

to evaluate features individually or in a pairwise manner. The new function has the advantage in that it can easily be applied to both continuous numerical and nominal features. It can also work on data sets with mixed numerical and nominal features.

In the experiments the PDSF–based feature selection was compared to the pairwise feature selection process. For both methods classification error and execution time was measured. The PDSF–based feature selection proved to be much faster than the PFS. Based on the results of the experiments it can be assumed with a very high statistical significance that the PDSF–based feature selection yields classification results not worse than the exhaustive pairwise search. Therefore, the new approach can be used to significantly speed up computations without noticeable loss in classification accuracy.

Additional tests were performed on the same data sets to compare the probabilistic-dependence similarity function (PDSF) with correlation–based criterion proposed in our previous work. Results obtained in these tests suggest that the new criterion performs better on data sets with nominal and mixed features than the correlation–based approach.

Results presented in this paper as well as results included in our previous work (Michalak and Kwasnicka 2006a;b) suggest that a significant improvement in computation speed can be achieved without any loss in classification accuracy by selecting features individually or in pairwise manner according to the assessed level of dependence between features. Future research may include extending the presented feature selection method to more than two–variable sets. Such an extended method would require a measure of feature dependence level in feature sets containing three and more elements. The measure of feature dependence level could then be used to decide whether feature pairs should be extended to sets containing three and more features.

References

Blake, C., Newman, D., Hettich, S., and Merz, C. 1998. Uci repository of machine learning databases.

Choi, J. H., Kim, K. L., Huh, W., Kim, B., Byun, J., Suh, W., Sung, J., Jeon, E. S., Oh, H. Y., and Kim, D. K. 2004. Decreased number and impaired angiogenic function of endothelial progenitor cells

in patients with chronic renal failure. *Arterioscl. Thromb. Vasc. Biol.*, 7(24):1246–1252.

Cover, T. and Van Campenhout, J. 1976. On the possible orderings in the measurement selection problem. In *Proceedings 3rd International Joint Conference on Pattern Recognition*, pages 245–248, Coronado. IEEE.

Das, S. 2001. Filters, wrappers and a boosting-based hybrid for feature selection. In *ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning*, pages 74–81, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Duda, R. O., Hart, P. E., and Stork, D. G. 2000. *Pattern Classification (2nd Edition)*. Wiley-Interscience.

Gasca, E., Sanchez, J., and Alonso, R. 2006. Eliminating redundancy and irrelevance using a new mlp-based feature selection method. *Pattern Recognition*, 39:313–315.

Hall, M. A. 1998. Correlation-based feature selection for machine learning. Technical report, The University of Waikato, Department of Computer Science, Hamilton, New Zealand.

Harol, A., Lai, C., Pekalska, E., and Duin, R. P. W. 2007. Pairwise feature evaluation for constructing reduced representations. *Pattern Analysis and Applications*, 10(1):55–68.

Kittler, J. 1978. *An Introduction to Feature Extraction*, pages 41–60. Sijhoff and Noordhoff, the Netherlands.

Kohavi, R. and John, G. H. 1997. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1–2):273–324.

Kwasnicka, H. and Orski, P. 2004. Genetic algorithm as an attributes selection tool for learning algorithms. *Intelligent Information Systems 2004, New Trends in Intelligent Information Processing and Web Mining Proceedings of the International IIS: IIP WM04 Conference*, pages 449–453.

Liu, Y. and Schumann, M. 2005. Data mining feature selection for credit scoring models. *Journal of the Operational Research Society*, 56:1099–1108.

Masaeli, M., Fung, G., and Dy, J. 2010. From transformation-based dimensionality reduction to feature selection. In *International Conference on Machine Learning*.

Michalak, K. and Kwasnicka, H. 2006a. Correlation-based feature selection strategy in classification problems. *Applied Mathematics and Computer Science*, 16(4):503–511.

Michalak, K. and Kwasnicka, H. 2006b. Correlation-based feature selection strategy in neural classification. In *ISDA '06: Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications (ISDA'06)*, pages 741–746, Washington, DC, USA. IEEE Computer Society.

Moller, M. F. 1993. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*, 6:525–533.

Pekalska, E., Harol, A., Lai, C., and Duin, R. P. W. 2005. Pairwise selection of features and prototypes. In *Computer Recognition Systems, Proceedings of the 4th International Conference on Computer Recognition Systems, CORES'05, May 22-25, 2005, Rydzyna Castle, Poland*, volume 30 of *Advances in Soft Computing*, pages 271–278. Springer.

Pudil, P., Novovicova, J., and Kittler, J. 1994. Floating search methods in feature selection. *Pattern Recogn. Lett.*, 15(11):1119–1125.

Quinlan, J. R. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers.

Verleysen, M. 2003. Learning high-dimensional data. In Ablameyko, S., Goras, L., Gori, M., and Piuri, V., editors, *Limitations and Future Trends in Neural Computation*. IOS Press.

Xing, E. P., Jordan, M. I., and Karp, R. M. 2001. Feature selection for high-dimensional genomic microarray data. In *Proc. 18th International Conf. on Machine Learning*, pages 601–608, San Francisco, CA, USA. Morgan Kaufmann.

Yang, J. and Honavar, V. 1998. Feature subset selection using a genetic algorithm. *IEEE Intelligent*

Systems, 13:44–49.