Mining Direct Marketing Data by Ensembles of Weak Learners and Rough Set Methods

Jerzy Błaszczyński[1], Krzysztof Dembczyński[1], Wojciech Kotłowski[1], and Mariusz Pawłowski[2]

[1] Institute of Computing Science, Poznań University of Technology, 60-965 Poznań, Poland
{jblaszczynski, kdembczynski, wkotlowski}@cs.put.poznan.pl
[2] Acxiom Polska, 02-672 Warszawa, Poland
mariusz.pawlowski@acxiom.com

Abstract. This paper describes problem of prediction that is based on direct marketing data coming from Nationwide Products and Services Questionnaire (NPSQ) prepared by Polish division of Acxiom Corporation. The problem that we analyze is stated as prediction of accessibility to Internet. Unit of the analysis corresponds to a group of individuals in certain age category living in a certain building located in Poland. We used several machine learning methods to build our prediction models. Particularly, we applied ensembles of weak learners and ModLEM algorithm that is based on rough set approach. Comparison of results generated by these methods is included in the paper. We also report some of problems that we encountered during the analysis.

1 Introduction

Direct marketing is one of the most popular form of promotion and selling. It is attractive, because its effectiveness can be measured, for example, by responses of customers to the promotion. Database of profiled customers is an important element in this type of marketing. From this database, one can select customers that with high probability will response to the promotion. To perform such a selection of customers one needs a prediction model. This model is derived from a sample of customers that are relatively well-known, for example, customers that fulfilled a special type of a questionnaire. Only attributes that are easily achieved for out-of-sample customers are used as predictors in the model. Acxiom is a company which aims in direct marketing technologies and is focused on integration data, services and technology to create innovative solutions that improve customer relationships. The mission of the company is to transform data collected from different sources (such as questionnaires, official registries) into marketing, actionable information, which helps to understand customer preferences, predict their behavior and increase effectiveness of direct marketing campaigns.

The problem considered in this paper consists in prediction of accessibility to Internet. Unit of the analysis corresponds to a group of individuals in certain age category living in a certain building located in Poland. The information about access to Internet comes from Nationwide Products and Services Questionnaire (NPSQ) prepared by Polish division of Acxiom Corporation. Predictors are taken from different databases coming from Acxiom and official registries that we further describe in Section 2. From the business perspecitive, the most important factors indicating quality of constructed prediction models are precision and true positive ratios. The preliminary analysis has shown that the problem is hard and any small improvement of these ratios in comparison to the random classifier that takes into account distribution of classes will be acceptable.

We have used several machine learning methods to build our prediction models. Particularly, we have applied algorithms constructing ensembles of weak learners and ModLEM rule induction algorithm that is based on rough set approach. Ensembles of weak learners, sometimes called decision committees, have been successfully used to many real-world problems. Some of these methods are treated today as off-the-shelf methods-of-choice. Rough set approach has also proved to be a useful tool for solving classification problems. It allows to determine inconsistencies between analyzed objects (such as we have found in the analyzed database) and functional dependencies between subsets of attributes. Comparison of models built using these methods is included in the paper. First results of our analysis were very promising. However, due to a mistake made in the preparation of data, these results were overestimated. Results that we obtained on the fixed data are worse, but still acceptable.

The paper is organized as follows. In Section 2, the problem is formulated from business and machine learning perspectives. Data sources and project database schema is also described there. Section 3 includes general description of prediction methods used in the analysis. In Section 4, we describe problems faced during the analysis and show results of predictions. The last section concludes the paper.

2 Problem Statement

Acxiom in Poland collects data from many different sources, aggregated on different levels: individual person, household, building, statistical region, community. Since 1999 Acxiom is continuously running nationwide survey collecting information thru over 100 questions about lifestyle, shopping behavior, product and services preferences and also demographic profile of households and individuals. Survey database (NPSQ) consist of over 2.2 million household and is the biggest and the most comprehensive database of this kind. Even thou, this database cover only about 15% of all households in Poland. The challenge at this point is how to generate precise information about rest of the population. Collecting data thru survey would take long time, a lot of money with no guaranty that this project will ever succeed. One of the solutions is to look at the data available on market, integrate and analyze it and transform into information we are looking for. In the project described in this paper following data sources has been used:

– Database of buildings from registry of inhabitants (PESEL), which consist of data about over 5 millions of buildings in Poland with information of number of flats, inhabitants, age and sex structure,
– Database of statistical regions with demographic data (Acxiom),
– Regional Data Bank with information aggregated on community level including wide range of information about local markets (GUS BDR).

Aim of the project was to "translate" information from survey to the lowest possible level of aggregation based on data from sources listed above. The first step in this project was to define the most precise unit of analysis. Having data from PESEL database with age structure assigned to each building we decided to define unit of analysis as an age category within each building. This definition is the closest level of data aggregation to the level of individual person and thus allows describing individuals in the most precise way. This definition causes a simplification that each person, living under certain building in certain age brackets will be assigned the same characteristics mined out of all of the data sources used in this project. However, results of initial analysis shows that homogeneity of groups of individuals defined in this way is acceptable. We have assumed that the prediction model will be deterministic. It means that outputs of the model will indicate if a group of individuals in a certain age category living in a certain building has access to Internet or not. Alternatively, we could use a probabilistic model (i.e., in such a case outputs of the model will indicate distribution of access to Internet for the unit of analysis).

After defining basic unit of analysis the next task was to integrate all data sources. This process was simplified thanks to having complex reference database, which includes all relations between addresses, statistical regions and communities and also having the same standard of writing street names in PESEL and NPSQ database. Finally, after integration each record in PESEL database was assigned record id from NPSQ, Acxiom and GUS BDR databases. Combining databases thru joined id's allows building flat table including all data from all of the sources assigned to the final level of analysis.

The database used in the case study contains more than 200 000 records and totally 751 attributes (without counting key and auxiliary attributes). The database after integration process was transformed for analysis purposes to a model that is similar to star schema well-known in dimensional modelling. In our schema, fact table contains attributes taken from NPSQ database, dimensions are tables from PESEL, Acxiom and GUS BDR databases. The model is presented in Figure 1. Such a construction of the database improves performance and facilitates the analysis. For example, when we want to analyze impact of attributes from GUS BDR's dimension on accessibility to Internet, it is enough to join the fact table with this dimension, omitting all other data.

Let us define the prediction problem in the formal way. Concluding the above, the aim is to predict the unknown value of an attribute y (sometimes called *output*, *response variable* or *decision attribute*) that represents accessibility to Internet of individual person using the known joint values of other attributes (sometimes called *predictors*, *condition attributes* or *independent variables*) $\mathbf{x} =$

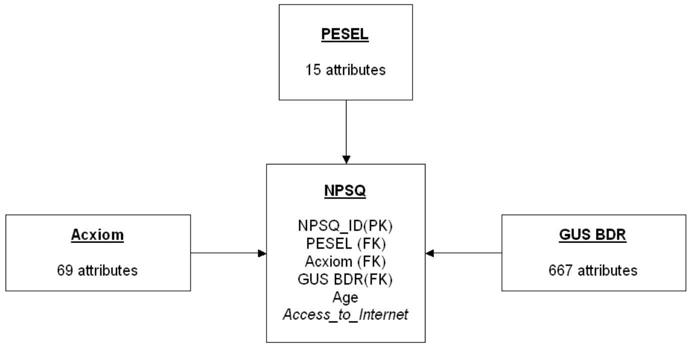

Fig. 1. Database schema used in case study.

$(x_1, x_2, \ldots, x_n)$. The goal of a learning task is to produce a function $F(\mathbf{x})$ from a set of training examples (or objects) $\{\mathbf{x_i}, y_i\}_1^N$ that predicts accurately y. Each training example corresponds to a responder of NPSQ, or, in other words, to a single record in NPSQ database. In the considered problem $y \in \{-1, 1\}$ indicates whether individual person has not or has access to Internet, respectively. In other words, all objects for which $y = -1$ constitute the decision class of individuals without Internet access, and all object for which $y = 1$ constitute the decision class of individuals with Internet access. These classes are denoted by Cl_{-1} and Cl_1, respectively. Condition attributes $\mathbf{x}$ refer, however, to the unit taken for the analysis, i.e., groups of individuals in a certain age category within certain building, because this information is easily achieved for individual persons that are not included in NPSQ database. The optimal classification procedure is given by:

$$F^*(\mathbf{x}) = \arg\min_{F(\mathbf{x})} E_{xy} L(y, F(\mathbf{x})) \tag{1}$$

where the expected value is over joint distribution of all attributes $(\mathbf{x}, y)$ for the data to be predicted. $L(y, F(\mathbf{x}))$ is loss or cost for predicting $F(\mathbf{x})$ when the actual value is y. The typical loss in classification tasks is:

$$L(y, F(\mathbf{x})) = \begin{cases} 0 & y = F(\mathbf{x}), \\ 1 & y \neq F(\mathbf{x}). \end{cases} \tag{2}$$

The learning procedure tries to construct $F(\mathbf{x})$ to be the best possible approximation of $F^*(\mathbf{x})$. The prediction model based on $F(\mathbf{x})$ is then applied to individual person described by attributes $\mathbf{x}$ referring, however, to certain age category within certain building to get information about her/his access to Internet.

3 Ensembles of Weak Learners and Rough Set Methods

To solve the defined problem we have used two types of algorithms: ensembles of weak learners (sometimes called decision committees) and rough set methods.

Algorithm 1: Ensemble of Weak Learners [4]

input : set of training examples $\{\mathbf{x}_i, y_i\}_1^N$
M – number of weak learners to be generated.
output: ensemble of weak learners $\{f_m(\mathbf{x})\}_1^M$.
$F_0(\mathbf{x}) = \arg\min_{\alpha\in\Re} \sum_i^N L(y_i, \alpha)$;
for $m = 1$ *to* M **do**
  $\mathbf{p} = \arg\min_{\mathbf{p}} \sum_{i\in S_m(\eta)} L(y_i, F_{m-1}(\mathbf{x}_i) + f(\mathbf{x}_i, \mathbf{p}))$;
  $f_m(\mathbf{x}) = f(\mathbf{x}, \mathbf{p})$;
  $F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \nu \cdot f_m(\mathbf{x})$;
end
ensemble $= \{f_m(\mathbf{x})\}_1^M$;

The first algorithm forms an ensemble of subsidiary classifiers that are simple learning and classification procedures often referred to as weak learners. The ensemble members are applied to classification task and their individual outputs are then aggregated to one output of the whole ensemble. The aggregation is computed as a linear combination of outputs or a simple majority vote. The most popular methods that are used as weak learners are decision tree induction procedures, for example C4.5 [7] or CART [3]. There are several approaches to create ensembles of weak learners. The most popular are bagging [2] and boosting [9]. In [4], Friedman and Popescu have formulated a general schema of algorithm that can simulate these two approaches. The schema is presented as Algorithm 1. In this procedure, $L(y_i, F(\mathbf{x}_i))$ is a loss function, $f_m(\mathbf{x}_i, \mathbf{p})$ is the weak learner characterized by a set of parameters $\mathbf{p}$ and M is a number of weak learners to be generated. $S_m(\eta)$ represents a different subsample of size $\eta \le N$ randomly drawn with or without replacement from the original training data. ν is so called "shrinkage" parameter, usually $0 \le \nu \le 1$. Values of ν determine the degree to which previously generated weak learners $f_k(\mathbf{x}, \mathbf{p})$, $k = 1..m$, effect the generation of a successive one in the sequence, i.e., $f_{m+1}(\mathbf{x}, \mathbf{p})$.

Classification procedure is performed according to:

$$F(\mathbf{x}) = sign(a_0 + \sum_{m=1}^{M} a_m f_m(\mathbf{x}, \mathbf{p})). \tag{3}$$

$F(\mathbf{x})$ is a linear classifier in a very high dimensional space of derived variables that are highly nonlinear functions of the original predictors $\mathbf{x}$. These functions are induced by weak learners, for example, they are decision trees. Parameters $\{a_m\}_0^M$ can be obtained in many ways. For example, they can be set to fixed values (for example, a_0=0 and $\{a_m = 1/M\}_1^M$), computed by some optimization techniques, fitted in cross-validation experiments or estimated in the process of constructing an ensemble (like in AdaBoost [9]).

According to Friedman and Popescu [4], bagging method [2] may be represented by Algorithm 1 and classification procedure (3) by setting $\nu = 0$, subsamples $S_m(\eta)$ are drawn randomly with replacement, where η is given by a user, $a_0 = 0$ and $\{a_m = 1/M\}_0^M$. AdaBoost uses exponential loss, $L(y, F(\mathbf{x})) = exp(-y \cdot F(\mathbf{x}))$, for $y \in \{-1, 1\}$, and corresponds to Algorithm 1 by setting $\nu = 1$ and $S_m(\eta)$ to be a whole set of training examples.

ModLEM [5] is a rule induction procedure that is based on rough set approach [6]. Decision rules are simple logical statements of a form: "*if* [conditions], *then* [decision]". The induction of rules in rough set approach consists of the two following phases: calculation of lower and upper approximations of decision classes, and induction of certain rules from lower approximations and possible rules from upper approximations. The first phase is useful to show inconsistencies in the data. Inconsistencies that we consider arise when objects with the same values of condition attributes are assigned to different decision classes. Lower approximation of a class is composed of all its objects that are consistent. Upper approximation holds also inconsistent objects. In the second phase, calculated approximations are used in rule induction process to obtain rules that represent certain knowledge (i.e., certain rules) and rules that represent possible knowledge (i.e., possible rules). In our problem, we expected that further insight into inconsistencies in groups of individuals that define units of analysis will allow us to obtain more precise classification results.

ModLEM is a specialized version of a general procedure that is known as *sequential covering*, very often used in rule induction systems. In fact, this procedure can be presented (see Algorithm 2) in a similar manner to Algorithm 1. One can remark on this basis that a set of decision rules may be then treated as an ensemble of decision rules that are very simple classifiers. Let us notice that Friedman and Popescu [4] has recently also developed a variant of Algorithm 1 that constructs an ensemble of decision rules. These rules are created in a specific way from a decision tree induced in each iteration of the algorithm. In sequential covering procedure, positive and negative examples are distinguished. Rules are built in such a way that they cover only positive examples. For certain rules assigning examples to a given class Cl_i, $i \in \{-1, 1\}$ positive examples are those from lower approximation of this class. Analogously, positive examples for possible rules are those from upper approximation of this class. A set of positive examples is denoted by $\hat{X}$. A rule parameterized by $\mathbf{c}$ is defined as:

$$f(\mathbf{x}, \mathbf{c}) = \begin{cases} 1 & \text{if } \mathbf{x} \text{ is covered by conditions } \mathbf{c} \text{ and rule assigns to } Cl_1, \\ 0 & \text{if } \mathbf{x} \text{ is not covered by conditions } \mathbf{c}, \\ -1 & \text{if } \mathbf{x} \text{ is covered by conditions } \mathbf{c} \text{ and rule assigns to } Cl_{-1}. \end{cases} \tag{4}$$

Loss function is defined as:

$$L(y, F_m(\mathbf{x})) = \begin{cases} 0 & y = sign(F_m(\mathbf{x})), \\ 1 & y \neq sign(F_m(\mathbf{x})). \end{cases} \tag{5}$$

Procedure of constructing a decision rule consists in a greedy heuristic that minimize $\sum_{i \in \hat{X}} L(y_i, F_m(\mathbf{x}_i) + f(\mathbf{x}_i, \mathbf{c}))$.

Classification procedure is performed according to a distance-based version of the bucket brigade algorithm [1]. The decision to which class the classified object is assigned depends on three factors: *strength* (str), *specificity* (spe) and

Algorithm 2: Sequential covering

input : set of training examples $X = \{\mathbf{x}_i, y_i\}_1^N$
set of positive examples $\hat{X} \subset X$.
output: set of rules $\{f_m(\mathbf{x})\}_1^M$.
$m = 1$;
while $\sum_{i \in \hat{X}} L(y_i, f(\mathbf{x}_i, \mathbf{c})) \neq 0$ **do**
  $c = \arg\min_{\mathbf{c}} \sum_{i \in \hat{X}} L(y_i, F_m(\mathbf{x}_i) + f(\mathbf{x}_i, \mathbf{c}))$;
  $f_m(\mathbf{x}) = f(\mathbf{x}, \mathbf{c})$;
  $F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + f_m(\mathbf{x})$;
  $m = m + 1$;
end
$M = m$; $rules = \{f_m(\mathbf{x})\}_1^M$;

matching (*mat*) factor. All those factors are computed for rule $f(\mathbf{x}, \mathbf{c})$. *Strength* is a total number of training examples classified correctly by the rule. *Specificity* is number of conditions of the rule. *Matching* factor reflects number of selectors of the rule matched by object x.

$$F(\mathbf{x}) = \arg\max_y \sum_{f(\mathbf{x},\mathbf{c})=y} str(f(\mathbf{x}, \mathbf{c})) \cdot spe(f(\mathbf{x}, \mathbf{c})) \cdot mat(f(\mathbf{x}, \mathbf{c})). \tag{6}$$

For more detailed description of ModLEM algorithm refer to [5].

4 Problems Indicated and Experimental Results

The first problem, that we encountered, was the size of the database. It contained over 200 000 examples, described by 751 attributes. To succeed with our analysis we decreased the number of attributes using filter method based on information gain criterion [7]. We have sorted all attributes with respect to this criterion and chosen 139 attributes, for which the value of information gain was on acceptable level. Unfortunately, majority of attributes has this value very close to zero and the highest value was also very small, what shows that the problem is very hard. To make learning process more efficient, we decided to divide the database into smaller data bins. We have decided to split the database with respect to values of one of the attributes. The chosen attribute is ”the number of individuals in a building” denoted as R later in this paper. We decided to use this attribute since we expected to obtain different prediction models, depending on the type of the building considered (e.g. family houses, blocks of flats, etc.). The database was then split into 21 bins of similar size, but with different consecutive values of R in each bin. To check impact of this type of splitting on the resulting models, we have also randomly splited the database into 20 bins of equal size. Then, we have compared accuracy of models resulting from these two splittings. The results of the comparison are presented later in the paper.

For purpose of our experiments, the database has been divided into two balanced sets. The one that is used to train and to validate prediction models. The

Table 1. First results obtained in the case study on test file.

Classifier	Bagging with j48			ModLEM		
	Class	True positive	Precision	Class	True positive	Precision
	-1	0.843	0.868	-1	0.844	0.871
	1	0.463	0.409	1	0.451	0.4

second is a test set used in a final verification of built models. Computations were performed using Weka package [11] for ensembles methods and ROSE program [8] for ModLEM. The problem of prediction of accessibility to Internet is imbalanced. Only approximately 20% of NPSQ records indicate access to Internet. To deal with it, we have used CostSensitiveClassifier in each computations performed in Weka. ModLEM algorithm is less prone to imbalancement of data sets and it performed without cost sensitive adaptation.

From the business perspective, the most important factors indicating quality of models are precision and true positive ratios. They are defined for each decision class Cl as follows:

$$precision(Cl) = \frac{|\text{set of examples correctly classified to } Cl|}{|\text{set of all examples classified to } Cl|},$$
$$true\_positive(Cl) = \frac{|\text{set of examples correctly classified to } Cl|}{|\text{set of all examples from } Cl|},$$

where $|A|$ denotes a number of examples in set A.

To get the idea about improvement of constructed models, one can compare them to random classifier that takes into account distribution of classes. In our problem, where we have two decision classes, Cl_{-1} and Cl_1, it is easy to estimate the probability of error for such a random classifier. In our case 80% of examples belong to class Cl_{-1}, 20% to class Cl_1. Random algorithm would classify correctly 68% of objects (64% from class Cl_{-1} and 4% from class Cl_1). The precision in class Cl_1 would be 20%, and the true positive ratio would be also 20%, etc. While analyzing hard problems as it is in our case, we do not expect to get high improvement of precision and true positive ratios as compared to random classifier. Usually, even small improvements are acceptable. In our problem, we expected that improvement around 10 percent points would be a good result.

First experiments shown very promising results. We have obtained the best results on the test set using ModLEM algorithm and Bagging ($\eta = N$) with j48 that is Weka implementation of C4.5 [7]. These results are presented in Table 1. The parameters of j48 and ModLEM were fitted in cross-validation experiments, and it was striking for us that the best models were obtained using parameters that causes decision trees to be very detailed and high number of long decision rules. Unfortunately, the results presented in Table 1 are overestimated. It was caused by a mistake that we have made in preparation of data for the experiment.

Table 2. Revised results obtained in the case study.

Classifier	Bagging with j48			AdaBoost with DS		
Type of split	Class	True positive	Precision	Class	True positive	Precision
Random	-1	0.596	0.852	-1	0.579	0.851
	1	0.582	0.262	1	0.59	0.258
by R	-1	0.595	0.855	-1	0.591	0.85
	1	0.591	0.265	1	0.577	0.258
for $R < 4$	-1	0.555	0.875	-1	0.556	0.875
	1	0.716	0.31	1	0.717	0.311
test set,	-1	0.574	0.846	-1	0.569	0.846
split by R	1	0.614	0.281	1	0.618	0.280
Classifier	Bagging with SVM			ModLEM		
Random	-1	0.591	0.855	-1	0.899	0.808
	1	0.593	0.265	1	0.141	0.259
by R	-1	0.597	0.858	-1	0.829	0.818
	1	0.599	0.268	1	0.235	0.248
for $R < 4$	-1	0.529	0.873	-1	0.869	0.797
	1	0.725	0.301	1	0.226	0.333
test set,	-1	0.574	0.848	-1	0.777	0.808
split by R	1	0.621	0.283	1	0.323	0.284

This mistake consists in presence of duplicated record in NPSQ. In some cases there were two records for a household for which there were no difference on condition and decision attributes. In many cases, one record from such a pair was placed in the training set, and other record from such a pair was placed in the test set. In total there were 17% of such redundant records.

Results on fixed data are much worse, but still acceptable. The algorithms that performed best are: bagging ($\eta = N$) with j48, bagging ($\eta = N/10$) with linear Support Vector Machines (SVM) [10], AdaBoost with decision stumps (DS) (i.e., one level decision trees). Results obtained by ModLEM are worse. We expect that it is caused by higher imbalance between decision classes and increased overall level of inconsistencies between examples for which R was high. We expect also that an approach to classification that is more focused on objects from areas of inconsistency that are detected by rough set approach will provide better results. Table 2 contains detailed results of experiments. These results come from 10-fold cross-validation for each bin, averaging these results over bins. In cross-validation experiments we have used two types of splitting. We present also results for the bin in which R is lower than 4. Finally, we present results on test set, where split by R was used.

We used t-test to check whether the constructed models increase significantly precision and true positive ratios. These tests shown that there are significant improvements between results of models as compared to results achieved by random classifier. When it comes to check, whether splitting with respect to R

impacts accuracy of predicted models, there are almost no differences between this type of splitting and random split. Division with respect to R parameter does not influence considerably overall values of these factors. Let us underline that usage of this type of splitting gives further insight into the problem. For bins with small R, the results are better than for the whole data set. It is worth noting that in the case of data with duplicates, there was a large difference of model quality factors between these two types of splitting. This difference is in favor of models built on bins created with respect to R.

5 Conclusions

In this paper we have described project that concerns a real-world problem of mining direct marketing data. We have applied several machine learning methods to predict accessibility to Internet. When solving such problems one should be very careful at initial stages of preparing data for the experiment. Mistake that was made at this stage lead us to overestimation of obtained results. The results that we have obtained after correction of the prepared data are worse but still acceptable. The best results were obtained by application of ensembles of weak learners. There is slight advantage of bagging with linear SVM, where subsamples were of size $N/10$. In our opinion, the results of ModLEM can be improved if we apply more sophisticated strategy for objects from areas of inconsistency that are detected by rough set approach. It is included in our further research plans.

References

1. Booker, L. B., Goldberg, D. E., Holland, J. F.: Classifier systems and genetic algorithms. In Carbonell, J. G. (ed.): Machine Learning. Paradigms and Methods. Cambridge, MA: The MIT Press, (1990) 235–282
2. Breiman, L.: Bagging Predictors. Machine Learning 2 **24** (1996) 123-140
3. Breiman, L., Friedman, J. H., Olshen, R. A., Stone, C. J.: Classification and Regression Trees. Wadsworth (1984)
4. Friedman, J. H., Popescu, B. E.: Predictive Learning via Rule Ensembles. Research Report, Stanford University, `http://www-stat.stanford.edu/~jhf/` (last access: 1.06.2006), February (2005)
5. Grzymala-Busse, J. W., Stefanowski, J.: Three discretization methods for rule induction. International Journal of Intelligent Systems 1 **16** (2001) 29–38
6. Pawlak, Z.: Rough Sets. Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1991)
7. Quinlan, J. R.: C4.5: Programs for Machine Learning, Morgan Kaufmann (1993)
8. Rough Sets Data Explorer (ROSE2), `http://idss.cs.put.poznan.pl/site/rose.html` (last access: 1.06.2006)
9. Schapire, R. E., Freund, Y., Bartlett, P, Lee, W. E.: Boosting the margin: A new explanation for the effectiveness of voting methods. The Annals of Statistics 5 **26** (1998) 1651–1686
10. Vapnik, V.: The Nature of Statistical Learning Theory. Springer-Verlag, New York (1995)
11. Witten, I., H., Frank, E.: Data Mining: Practical machine learning tools and techniques, 2nd Edition. Morgan Kaufmann, San Francisco (2005)