An experimental evaluation of improving rule based classifiers with two approaches that change representations of learning examples

Jerzy Stefanowski

Institute of Computing Science, Poznań University of Technology, ul. Piotrowo 3A, 60-965 Poznań, Poland

Abstract

Improving the classification performance of decision rules induced from examples is discussed in this paper. As standard rule based classifiers may not work well enough for some complex learning tasks, we consider an extension of the rule induction algorithm MODLEM with two approaches that change the original representation of the learning examples. The first approach selects the subset of the most relevant attributes with the wrapper model while the other modifies the presence of some learning examples in the data by the bagging technique. The aim of this study is to experimentally evaluate on several data sets how these two different approaches improve the classification performance of the rule sets induced by the MODLEM algorithm.

Key words: Machine learning, rule induction, classification, attribute selection, multiple classifiers, the bagging approach, the MODLEM algorithm

1 Introduction

The aim of machine learning is to construct systems that automatically improve their performance with analysing experience represented by learning examples. In recent years many successful applications of machine learning have been developed, in particular in the domain of *knowledge discovery* [19,13]. One of common tasks considered in this context is the *prediction of object classification.* It consists of assigning a decision class label to a set of unclassified objects described by a fixed set of *attributes* (features, characteristics).

Email address: `e-mail: Jerzy.Stefanowski@cs.put.poznan.pl` (Jerzy Stefanowski).

Learning algorithms could be used to induce various forms of the representation of the classification knowledge from learning examples, e.g. decision trees, rules, Bayesian classifiers. In this paper we discuss a case, where the classification knowledge is expressed in a form of *decision rules.* The rules are represented as logical expressions of the following form:

IF (*conditions*) THEN(*decision class*),

where conditions are formed as a conjunction of elementary tests on values of attributes and a decision part of the rule indicates the assignment of an object, which satisfies the condition part, to the given decision class. A number of various algorithms have already been developed to induce such rules (for some reviews the reader can consult, e.g., [10,19,24,13]).

Let us remind that rules are one of the most popular type of knowledge representations used in practice. One of the main reasons for their wide application is their expressive and easily human-readable representation, see e.g. discussions in [19,13]. They were also successfully used in several applications. The interesting review paper on this topic was written by Simon and Langley [18]. Let us repeat their opinion saying that "machine learning aims to provide increasing level of automation of knowledge engineering process (...) with automatic techniques that improve accuracy or efficiency by discovering and exploiting regularities (in data). The ultimate test of machine learning is its ability to produce systems that are used regularly industry, education and elsewhere" [18]. In their study they described several cases where decision rules or trees were applied to real-life problems. For the reader of this journal the following problems could be interesting: increasing yield in chemical process control, diagnosis of mechanical devices, monitoring quality of rolling emulsions, reducing banding in rotogravure printing, analysis of breakdowns in electrical networks, inducing knowledge bases for diagnosing faults in circuit boards. One can have a look in other reviews on engineering applications, e.g. some chapters in [19,13] or in [16] (where it is presented how rules are used in such sub-fields as e.g. signal and image analysis, scheduling, material analysis, power system security analysis or environmental cases). Let us also notice a study of applying inductive rule learning to the mechanical engineering design [20]. Moreover, the author of the following paper took part in several projects, where the rule induction was used, e.g., for technical diagnostics of rolling bearings and other rotating machines [21], determining the maintenance policy for the fleet of buses on the basis of their exploitation characteristics [28], and supporting diagnostic decisions with knowledge derived from the medical data, see the reviews in [24]. The above examples constitute only a fraction of the fielded applications of rule induction in engineering and show the potential usefulness of these approaches.

When using decision rules for predicting aims, *classification* (predictive) *accuracy* plays a major role in an evaluation of rules. It is easy to understand why this is so - accuracy is a primary concern in main applications of supervised learning systems and is easily measured as opposed to, e.g., rule interestingness which is more subjective [22]. The classification accuracy is typically estimated in an experimental way by applying the rule set to classify *testing examples* and it is defined as the relative frequency of correct classifications to all tested examples [26]. Although different rule based classifiers have been proved to be efficient for several learning problems, they may not led to satisfactory classification accuracy for other, more complex and difficult, data sets. These difficulties may be caused, e.g., by existence of noisy examples or irrelevant attributes. Moreover, decision concepts may be too complex, non-linear and difficult to be learned by standard algorithms. One the approaches to improve the classification performance for such non-trivial data sets is to *change representation* of the input data in such a way that the learning system could better learn the classification problem in new dimensions. Let us remind that determining the appropriate representation space for learning, e.g., by choosing attribute relevant to the problem at hand, is a crucial issue while constructing learning systems [14,19]. Specific transformation methods looking for "better" representation space, possibly integrated with learning algorithms, could improve the learning process. In this paper, we will consider two different approaches to modify the representation space, which correspond to both "dimensions" of the original data description. These are:

(1) changing the set of attributes describing examples,
(2) changing the presence of some learning examples in the learning set, given a fixed set of originally predefined attributes.

In the first approach, it is checked whether removing some attributes could improve the classification performance of the rule classifier. We will use the method for selecting the most relevant attributes, which is based on the forward or backward stepwise search strategy applied inside, the so called, *wrapper model* originally described in [14].

In the second approach, the set of originally defined attributes remains unchanged while the presence of some learning examples is modified. It is performed by a specific sampling from the original training data considered within the framework of, so called, *multiple classifiers* [9,27]. In this paper we focus attention on the *bagging* approach [2], which manipulates the input data to get several different learning sets by using sampling with replacement of learning examples. Then, different classifiers are generated from these learning sets and combined by a voting strategy to form a composite classifier.

Although these both approaches are already known, it is interesting to see how they improve the performance of the rule based classifier. According to our best knowledge the rule classifier has not been considered yet together with these approaches. The bagging approach was mainly studied for decision trees, while the attribute selection mainly examined for instance based learning algorithms. The only exception is the previous preliminary author's study on using rule induction with the bagging [25], which gave encouraging results for the current study. Moreover, we can notice that the usefulness of these approaches has not been directly compared for the same learning algorithm.

The aim of this paper is to experimentally examine on a collection of different data sets the application of these approaches to a rule induction algorithm, called MODLEM. This algorithm has been previously introduced by the author in [23]. It is particularly well suited for analysing data containing a mixture of numerical and qualitative attributes. Although the previous comparative studies [7,12,24] showed the good performance of this algorithm on typical benchmark data sets, one cannot expect that it works also so efficiently for other more complex problems. Therefore, the main contribution of this paper is to verify how the both considered approaches could improve the classification performance of the rule classifier induced by MODLEM for more difficult data sets.

The paper is organised as follows. In Section 2, we begin with a presenting the MODLEM algorithm. Section 3 contains a brief description of the attribute selection method. Then, in Section 4, we describe modifications of learning sets by the bagging approach. Results of the comparative experiments are given in Section 5. The discussion of these results and conclusions are presented in the final section.

2 The rule induction by the MODLEM algorithm

The rule induction algorithm, called MODLEM, has been introduced by Stefanowski in [23], see also its formal descriptions in [24] or [12]. Due to the size of this paper we skip the more precise presentation of this algorithm and we only summarize its main idea. The algorithm is based on the scheme of a *sequential covering* and it heuristically generates a *minimal set* of decision rules for every decision concept (which is either decision class or its rough approximation in a case of inconsistent examples). Such a set of rules attempts to cover all (or the most significant) *positive examples* of the given concept and not to cover any *negative examples* (or as little as possible of them). The main procedure for rule induction scheme starts from creating a first rule by choosing sequentially the 'best' elementary conditions according to chosen criteria (i.e., the first candidate for the condition part is one elementary condition; If it does not fulfill the requirement to be accepted as a rule, then the next - currently best evaluated - elementary condition is added to the candidate for the condition part of the rule, etc.; This specialization is performed until the rule could be accepted). When the rule is stored, all learning positive examples that match this rule are removed from the current consideration. The process is iteratively repeated while some significant positive examples of the decision concept remain still uncovered. Then, the procedure is sequentially repeated for each set of examples from a succeeding decision concept. In the basic version of the MODLEM algorithm elementary conditions are evaluated by using one of two measures either *class entropy* or *Laplace accuracy* (see [5,24] for their definition and justification). It is also possible to consider a lexicographic order of two criteria measuring the rule positive cover and, then, its conditional probability (originally considered by Grzymala in his LEM2 algorithm or its last, quite interesting modification called MLEM2).

The extra property of the MODLEM algorithm is handling directly numerical attributes during rule induction while elementary conditions of rules are created, without any preliminary discretization phase [12]. In MODLEM elementary conditions are represented as either $(a < v_a)$ or $(a \geq v_a)$, where a denotes an attribute and v_a is its value. If the same attribute is chosen twice while building a single rule, one may also obtain the condition $(a = [v_1, v_2))$ that results from an intersection of two conditions $(a < v_2)$ and $(a \geq v_1)$ such that $v_1 < v_2$. For nominal attributes, these conditions are $(a = v_a)$. For more details about the function finding best elementary conditions see, e.g., [12,23].

Finally, the unordered set of induced rules is applied to classify examples using the classification strategy introduced by Grzymala in LERS system [11], which takes into account strength of all rules completely matched and also allows partially matches if no rule fits the description of the tested example.

3 The selection of attributes

For some problems not all attributes describing examples may be directly relevant to the object classification and some of them may be irrelevant or even redundant. If there are too many irrelevant attributes in the initial description of examples, the complexity of a learning process may increase. Moreover, they may decrease the classification accuracy of the induced classifier [14]. Thus, only the most relevant attributes should be used and various approaches were introduced to find the smallest subset of attributes leading to a higher, or similar, classification accuracy as obtained with all attributes.

The attribute selection can be considered as a search problem, where one selects a good attribute subset based on a selected evaluation measure [14,15]. Each state in the search space represents a subset of possible attributes. It is a partially ordered space, where each state has a child differing on exactly one attribute. Following the idea presented by Kohavi [14], to design a attribute selection algorithm one has to define three following components: *search algorithm* (technique that looks through the space of attribute subsets), *evaluation function* (used to evaluate examined subsets of attributes), and *classifier* (a learning algorithm that uses the final subset of attributes). These elements can be integrated in two ways. They are called *filter* and *wrapper* approaches, respectively [14]. In the filter approach, attributes are selected as a pre-processing step before a classifier is used. They are selected (i.e. filtered) basing on properties of data itself independently of the learning algorithm used as a classifier. The typical evaluation criteria come from statistics or information theory, e.g. Info-gain entropy, Chi-squared statistics or correlation based measures. On the other hand, in the wrapper approach, the search algorithm conducts a search for a good subset of attributes using the classifier itself as the evaluation function. An evaluation is usually done by estimating a classification accuracy obtained by this classifier.

Each filter and wrapper approach has its own strong and week points. Some authors claim [14] that the wrapper model is superior because it takes into account the bias of the learning algorithm in order to get a attribute subset. It should result in a better estimate of an accuracy on unseen data than the evaluation function used in the filter model whose, bias differs from that of the classifier. However, the major limitation of the wrapper approach is the additional computational cost resulting from using many times the cross-validation technique evaluating learning algorithm to check each attribute subset. On the other hand, the filter approach is less demanding. As to the wrapper approach one should remember that the classification accuracy for the final classifier induced using the selected subset of attributes should be evaluated on extra verification examples, which are not used in the learning phase to select the attribute subset. It is necessary to avoid the phenomena of the overfitting the learning algorithm to the examples [15].

The next issue concerns constructing the search algorithm within both models. Since the exhaustive search is of exponential complexity, it is more efficient to perform the heuristic search. Commonly employed algorithms are *backward elimination* and *forward selection.* Former starts with all attributes and successively removes the one that its elimination improves performance. The second starts with an empty set of attributes and successively adds the one with the best performance. There are also known more sophisticated variants of both approaches that combine performing either adding or removing operations as to get the best subset see, e.g.[17]. In our study we will use the wrapper model studying separately these two forward and backward search strategies.

4 The bagging approach to select learning examples

The bagging approach is one of the methods for creating multiple classifiers. By this name we understand a set of classifiers whose individual predictions (classification decisions) are combined into one classification decision in such a way that the system should improve the classification accuracy. Several different approaches to construct such systems have been already proposed, for some reviews see, e.g., [6,9,27]. The bagging belongs to homogeneous classifiers, where the original data sets are "slightly" modified to obtain diverse data sets. According to this idea the same learning algorithm is run several times, each time using a different distribution of the training examples. The generated classifiers are, then, combined to create a final classifier that is used to classify new objects. Except bagging, the other popular approach within this group is boosting [8]. Here, we chose the bagging as it is more directed to modifying the presence of examples in the learning sets. Moreover, we plan in future to integrate it together with the attribute selection.

The *Bagging* approach (**B**ootstrap **agg**regat**ing**) was introduced by Breiman [2]. It aggregates by voting classifiers generated from different bootstrap samples. The *bootstrap sample* is obtained by uniformly sampling objects from the training set with replacement. Each sample has the same size as the original set, however, some examples do not appear in it, while others may appear more than once. For a training set with m examples, the probability of an example being selected at least once is $1-(1-1/m)^m$. For a large m, this is about 1 - 1/e. Each bootstrap sample contains, on the average, 63.2% unique examples from the training set. Given the parameter T which is the number of repetitions, T bootstrap samples $S_1, S_2, \ldots, S_T$ are generated. From each sample S_i a classifier C_i is induced by the same algorithm and the final classifier C^* is formed by aggregating T classifiers. A final classification of object x is built by a uniform voting scheme on $C_1, C_2, \ldots, C_T$, i.e. is assigned to the class predicted most often by these sub-classifiers, with ties broken arbitrarily. The approach is presented briefly below. For more details see [2].

```
(input $LS$ learning set; $T$ number of bootstrap samples;
$LA$ learning algorithm
output $C^*$ classifier )
begin
    for $i = 1$ to $T$ do
    begin
        $S_i$ := bootstrap sample from $LS$; {sample with replacement}
        $C_i := LA(S_i)$; { generate a sub-classifier }
    end; { end for }
    $C^*(x) = \arg\max_{y \in K_j} \sum_{i=1}^{T} (C_i(x) = y)$
        {the most often predicted class}
end
```

Experimental results presented in [2,3] show a significant improvement of classification accuracy while using decision tree classifiers. For more theoretical discussion on the justification of bagging the reader is referred to [2].

Table 1
The characteristic of data sets used in the experimental evaluation

Data set	Number of examples	Number of classes	Number of attributes
bank	66	2	5
buses	76	2	8
zoo	101	7	8
hsv	122	4	11
iris	150	3	4
hepat	147	2	19
glass	214	6	9
bricks	216	2	10
vote	300	5	16
bupa	345	2	6
pima	768	2	8

5 Experiments

5.1 Conditions of experiments

The aim of the experimental study is to check how two considered techniques for modifying the representations of learning data, increase classification accuracy of the rule classifier induced by the MODLEM algorithm. More precisely, we compare the performance of:

(1) The single classifier obtained by MODLEM used for all attributes (without any selection).
(2) The single classifier induced also by MODLEM for selected subsets of attributes and all learning examples. Two search strategies inside the wrapper model are independently studied: forward selection and backward elimination (denotated as FS and BE, respectively).
(3) The bagging classifier composed of single classifiers induced by MODLEM with the complete set of attributes.

The MODLEM algorithm has been used with the entropy measure to choose elementary conditions. All experiments have been performed on the benchmark data sets, which are coming from Machine Learning Repository at the University of California at Irvine [1] or from the author's case studies (data *buses, hsv, bricks*, see [24]). Their characteristics is given in Table 1. The data

Table 2
The comparison of classification accuracies [%] for classifiers induced by using either all attributes or attribute subsets obtained by Forward Selection or Backward Elimination search strategies

Data set	All attributes	FS	BE
bank	93.81 ± 0.94	95.46 ± 1.1	92.15* ± 1.14
buses	97.20 ± 0.94	98.05* ± 1.15	97.38* ± 1.12
zoo	94.64 ± 0.67	95.01* ± 0.59	93.71* ± 0.59
hsv	54.52 ± 1.05	65.94 ± 0.69	58.41 ± 1.28
hepatitis	78.62 ± 0.93	83.91 ± 0.49	80.57 ± 0.79
iris	94.93 ± 0.5	94.67* ± 0.58	94.93* ± 0.5
glass	72.41 ± 1.23	71.42* ± 1.1	73.69* ± 1.04
bricks	90.32 ± 0.82	89.82* ± 0.37	89.89* ± 0.5
vote	92.67 ± 0.38	88.67 ± 0.82	93.91 ± 0.48
bupa	65.77 ± 0.6	62.28 ± 0.79	65.15* ± 0.55
pima	73.57 ± 0.67	74.92 ± 0.47	75.82 ± 0.51

were chosen taking into account their different properties (e.g. increasing the number of attributes, examples, decision classes; Moreover, some of these data are rather standard problems, easier to be learned, e.g. *iris* or *buses*, while others are more difficult ones as e.g. *hsv, bupa, pima*). The classification accuracy was estimated by a stratified version of 10-fold cross-validation technique, i.e. the training examples were partitioned into 10 equal-sized blocks with similar class distributions as in the original set.

5.2 The attribute selection

For all data sets we used the wrapper approach with two search strategies: forward selection (FS) or backward elimination (BE) . The obtained classification accuracies are presented in Table 2. They are calculated as an average classification accuracy with a standard deviation. An asterisk indicates that differences for the compared classifiers and the given data set are not statistically significant (it has been evaluated by two-paired t test with a confidence level α=0.05). Let us remind that the classifier obtained for a selected attribute subset should be evaluated on the verification examples, which are not used inside the wrapper internal search. Therefore, we employed two level "k-fold cross validation" technique. First, in the outside cross-validation, the examples are randomly divided into the learning and verification sets. Then,

Table 3
The comparison of classification accuracies [%] obtained by using the single MODLEM classifier and the bagging approach

Data set	Single classifier	Bagging
bank	93.81 ± 0.94	95.22 ±1.02
buses	97.20 ± 0.94	97.45* ± 1.13
zoo	94.64 ± 0.67	93.68 ± 0.70
hsv	54.52 ± 1.05	65.94 ± 0.69
hepatitis	78.62 ± 0.93	84.0 ± 0.49
iris	94.93 ± 0.5	94.33* ± 0.59
glass	72.41 ± 1.23	76.09 ± 0.68
bricks	90.32* ± 0.82	90.77* ± 0.72
vote	92.67 ± 0.38	96.01 ± 0.29
bupa	65.77 ± 0.6	75.69 ± 0.7
pima	73.57 ± 0.67	77.87 ± 0.39

for the wrapper search strategy, the other "inner"10-fold cross validation is used in the learning set to find the selected subset. In our experiments for the both cross validation techniques, *k* was equal to 10.

We have also analysed the structure of attribute subsets selected by both methods FS and BE. We have noticed that these subsets were not the same for any data set, except *glass* and *bupa*. Usually, the forward strategy selected much smaller number of attributes than the backward elimination.

5.3 The bagging modification of the learning set

While creating the multiple classifier, the parameter T being the number of sub-classifiers inside bagging was set at the following values: 3, 5, 7 and 10. Choosing these small values of T was inspired by good results obtained by Quinlan for studying C4.5 decision trees with the bagging [22] although Breiman originally used higher values.

The results of our experiments are given in Table 3. For each data set, the first column shows the average classification accuracy obtained by a single classifier over the ten cross-validations. The standard deviation is also given. The next column contains results for composite bagging classifier. An asterisk indicates that differences for compared classifiers and given data sets are not statistically significant. For nearly all data set we present results obtained for T=10 which was the best value among all tested, except: *bank* T=7, *hsv* T=5.

6 Discussion of results and final remarks

Let us start from discussing the results of the comparative experiments performed on 11 data sets. The classifier built with attributes determined by the forward selection strategy was better than the classifier using all attributes on 4 data sets (*bank, hsv, hepatitis, pima*) and worse on 2 data sets (*vote, bupa*). The difference between both classifiers was not significant on the remaining 5 data sets. Using backward elimination strategy improved classification accuracy on 4 data sets (*hsv, hepatitis, vote, pima*). For other data sets the differences between classifiers were not significant. Comparing the number of attributes selected by each search strategy, we noticed that the forward selection usually identified smaller subsets than the backward elimination. Moreover, the time of computations for the forward selection was usually shorter.

The other approach to modify the number of examples, i.e. the bagging, outperformed the single classifier on 7 of 11 data sets. The difference between classifiers were non-significant on 3 data sets (*buses, iris, bricks*) and the single classifier was better only for 1 data sets *zoo*. In fact, the slightly worse performance the bagging was observed for quite small data sets (e.g. *buses* - which could be difficult for representative sampling) or data sets rather easy to be learned by a standard single classifier (as e.g. *iris*). This is similar to observations from using decision tree algorithm inside the bagging [22]. Here we can also make a hypothesis that the bagging substantially improves the classification accuracy for data sets containing higher number of examples. Let us notice, that for some of these data bagging allowed to achieve the highest improvement of classification (see e.g. *glass, vote, bupa* or *pima*).

Comparing the experimental results between the two considered approaches we can conclude that the bagging approach is more efficient for improving the classification accuracy. It could be, somehow, explained by the specific properties of the rule induction algorithm. In general, the attribute selection should be useful for such learning algorithms, which are particularly sensitive to the presence of irrelevant attributes [15]. For instance, this could be the k–nearest neighbor algorithm, where all attributes influence the way of calculating the distance or similarity between examples. We have examined it in our previous studies [17]. However, the rule induction algorithm chooses the best elementary condition (referring to attributes) just inside a rule generation. Quite often, only part of the original attributes may be used in the finally induced rules. Therefore, this kind of learning algorithm has a kind of

”internal ability” to select the most relevant attributes. Although the usefulness of an additional attributes selection in the wrapper approach is limited, we observed an improvement for a few, rather more difficult, data sets, as *hsv, hepatitis*, etc., where some attributes seemed to be redundant.

On the other hand, the main aim of introducing the bagging approach was just to improve the classification performance [2]. Our experimental results have confirmed it also holds for using of the MODLEM algorithm. Let us remind that according to some authors [2,3] the bagging should work especially well for, the so called, unstable learning algorithms, i.e. ones whose output classifier undergoes major changes in response to small changes in learning data. Indeed the algorithm MODLEM is the unstable algorithm in this sense.

Finally, let us discuss the computational costs. As the bagging approach needs the number T of sub-classifiers, it requires around T more computational efforts than single learning algorithm. While the cost of the wrapper search depends on the number of attributes subsets to be examined during the search process. Moreover, for each examined subset it is necessary to perform the cross validation evaluation. Therefore, it is more demanding approach than the bagging. However, there is another disadvantage of the bagging approach - loosing a simple and easy interpretable structure of knowledge represented in a form decision rules. While, the attribute selection maintains a typical form of the single rule set.

For future research, it could be interesting to examine whether the integration of the bagging together with the attribute selection (inside the subsamples) could additionally improve the performance of the rule classifier.

Acknowledgment: The research has been supported from the KBN grant 3T11C 050 26.

References

[1] C. Blake, E. Koegh, C.J. Mertz, Repository of Machine Learning, University of California at Irvine 1999.

[2] L. Breiman, Bagging predictors. *Machine Learning*, **24** (2), (1996), 123–140

[3] E. Bauer, R. Kohavi, An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning* **36** (1/2), (1999), 105–139.

[4] K.J. Cios, W. Pedrycz, R.W. Swinarski *Data Mining Metods for Knowledge Discovery.* Kluwer, 1999.

[5] P. Clark, R. Boswell, Rule induction with CN2: some recent improvements. In: Kodratoff Y. (ed.) *Proceedings of 5th European Working Session on Learning - ESWL 91* Porto, Portugal, Springer Verlag, 1991, 151–163.

[6] T.G. Dietrich, Ensemble methods in machine learning. In: *Proc. of 1st Int. Workshop on Multiple Classifier Systems*, (2000) 1–15.

[7] M. Doumpos, C. Zopounidis, Rough sets based rule induction and multivariate statistical classification: a simulation study, *Computational Economics Journal*, 2002 (accepted).

[8] Y. Freund, R.E. Schapire, Experiments with a new boosting algorithm. In: *Proc. 13th Int. Conference on Machine Learning* (1996), 148–156.

[9] J. Gama, Combining classification algorithms. Ph.D. Thesis, University of Porto (1999).

[10] J.W. Grzymala-Busse, LERS - a system for learning from examples based on rough sets. In: Slowinski R. (ed.) *Intelligent Decision Support. Handbook of Applications and Advances of the Rough Sets Theory.* Kluwer, 1992, 3–18.

[11] J.W. Grzymala-Busse, Managing uncertainty in machine learning from examples. In: *Proc. 3rd Int. Symp. in Intelligent Systems*, Wigry, Poland, IPI PAN Press, 1994, 70–84.

[12] J.W. Grzymala-Busse, J. Stefanowski, Three approaches to numerical attribute discretization for rule induction. *International Journal of Intelligent Systems*, **16** (1), (2001), 29–38.

[13] W. Klosgen, J.M. Żytkow, *Handbook of Data Mining and Knowledge Discovery.* Oxford Press, 2002.

[14] G. John, R. Kohavi, K. Pfleger, Irrelevant features and the subset selection problem. *Proceedings of the Eleventh International Machine Learning Conference*, New Brunswick NJ, Morgan Kaufmann, 1994, 121-129.

[15] R. Kohavi, D. Sommerfield, Feature Subset Selection Using the Wrapper Method: Overfitting and Dynamic Search Space Topology. *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*, Montreal, AAAI Press, 1995, 192-197.

[16] J. Komorowski, Z. Pawlak, L. Polkowski, Skowron A., Rough Sets: tutorial. In: S.K. Pal, A. Skowron (eds.) *Rough Fuzzy Hybridization. A new trend in decision making*, Springer Verlag, Singapore, 1999, 3–98.

[17] K. Krawiec , J. Jelonek, J. Stefanowski, Comparative study of feature subset selection techniques for machine learning tasks. In: *Proceedings of VIIth Intelligent Information Systems* IIS'98 Malbork, 15-19 June 1998, IPI PAN Press Warszawa 1998, 68-77.

[18] P. Langley, H.A. Simon, Fielded applications of machine learning, In: R.S. Michalski, I. Bratko, M. Kubat (eds.), *Machine learning and data mining*, John Wiley & Sons, 1998, 113-129.

[19] R.S. Michalski, I. Bratko, M. Kubat (eds.), *Machine learning and data mining*, John Wiley & Sons, 1998.

[20] W. Moczulski, Inductive learning in design: A method and case study concerning design of antifriction bearing systems. In: R.S. Michalski, I. Bratko, M. Kubat (eds.), *Machine learning and data mining*, John Wiley & Sons, 1998, 203-220.

[21] R. Nowicki, R. Slowinski, J. Stefanowski, Rough sets analysis of diagnostic capacity of vibroacoustic symptoms. *Computers and Mathematics with Applications*, vol. 24 no. 7, 1992, 109-123.

[22] J.R. Quinlan, Bagging, boosting and C4.5. In: *Proceedings of the 13th National Conference on Artificial Intelligence*, 1996, 725–730.

[23] J. Stefanowski, The rough set based rule induction technique for classification problems. In: *Proceedings of 6th European Conference on Intelligent Techniques and Soft Computing* EUFIT'98, Aaachen 7-10 Sept. 1998, 109-113.

[24] J. Stefanowski, *Algorithims of rule induction for knowledge discovery.* (In Polish), Habilitation Thesis published as Series Rozprawy no. 361, Poznan Univeristy of Technology Press, Poznan, 2001.

[25] J. Stefanowski, Bagging and induction of decision rules. In: Klopotek M., Wierzchon S. Michalewicz M. (eds.), *Int. Symposium on Intelligent Systems*; Post-Proceedings of the IIS'2002. Series "Advances of Soft Computing", Physica Verlag, Heidelberg, 2002, 121-130.

[26] S.M. Weiss, C.A. Kulikowski, *Computer Systems That Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning and Expert Systems*, Morgan Kaufmann, San Francisco, 1991.

[27] G. Valentini, F. Masulli, Ensembles of learning machines. In: R. Tagliaferri and M. Marinaro (eds), *Neural Nets* WIRN Vietri-2002, Series Lecture Notes in Computer Sciences, Springer-Verlag, vol. 2486, 2002, 3-19.

[28] J. Zak, J. Stefanowski, Determining maintenance activities of motor vehicles using rough sets approach. In: *Proceedings of Euromaintenance'94 Conference*, Amsterdam, 1994, 39–42.