



**INSTYTUT BADAŃ SYSTEMOWYCH
POLSKIEJ AKADEMII NAUK**

TECHNIKI INFORMACYJNE TEORIA I ZASTOSOWANIA

Wybrane problemy
Tom 5 (17)

poprzednio

**ANALIZA SYSTEMOWA W FINANSACH
I ZARZĄDZANIU**

Pod redakcją
Andrzeja MYŚLIŃSKIEGO

Warszawa 2015



**INSTYTUT BADAŃ SYSTEMOWYCH
POLSKIEJ AKADEMII NAUK**

**TECHNIKI INFORMACYJNE
TEORIA I ZASTOSOWANIA**

Wybrane problemy
Tom 5 (17)

poprzednio

**ANALIZA SYSTEMOWA W FINANSACH
I ZARZĄDZANIU**

Pod redakcją
Andrzeja Myślińskiego

Warszawa 2015

Wykaz opiniodawców artykułów zamieszczonych
w niniejszym tomie:

Dr inż. Tatiana JAWORSKA

Dr inż. Jan OWSIŃSKI

Dr hab. inż. Andrzej MYŚLIŃSKI, prof. PAN

Prof. dr hab. inż. Piotr SIENKIEWICZ

Prof. dr hab. inż. Andrzej STRASZAK

Dr hab. Dominik ŚLĘZAK, prof. UW

Prof. dr hab. inż. Stanisław WALUKIEWICZ

Copyright © by Instytut Badań Systemowych PAN
Warszawa 2015

ISBN 83-894-7558-8

BOOSTING APPROXIMATE REDUCTS

Sebastian Widz

Systems Research Institute, Polish Academy of Sciences
PhD Studies, Warsaw
Poland
e-mail: widz@infovision.pl

Abstract. We consider classifiers ensembles constructed using boosting methodology. Each weak classifier is based on rough set inspired approach to deriving attribute subsets from data organized in a form of a decision system. We focus on approximate decision reducts calculated over universe of weighted objects. We show how methods derived in our previous research integrate smoothly with boosting approach. Two different approaches for approximate reduct calculation are analyzed. We discuss how our framework can be utilized for deriving meaningful attribute subset ensembles. Finally we test our methods on the benchmark data.

Keywords: Boosting, Classifier Ensembles, Rough Sets, Approximate Reducts, Feature Subset Selection

1 INTRODUCTION

Attribute selection plays an important role in knowledge discovery. It establishes the basis for more efficient classification, prediction and approximation models. Attribute selection methods originating from the theory of rough sets put a special attention on providing users with a better insight into data dependencies [26]. There are numerous rough set based algorithms aimed at searching for so called decision reducts – irreducible subsets of attributes that satisfy predefined criteria for keeping *enough* information about decision classes [1]. Those criteria reflect more or less explicitly a chance of misclassification of objects by if-then decision rules with their antecedents referring to the values of attributes and their consequents referring to decisions.

Original definition of a reduct is quite restrictive, requiring that it should determine decisions or, if data inconsistencies do not allow for full determinism, provide the same level of information about decisions as the complete set of attributes. However, there are a number of formulations of approximate reducts, which *almost* preserve original decision information [22]. Such approximate criteria usually rely on functions measuring

degrees of decision information induced by subsets of attributes and the corresponding thresholds specifying which subsets of attributes are *good enough*. Those degrees are usually computed from the training data, which makes the whole rough set based framework an example of a filter attribute selection approach [27].

Approximate decision reducts usually include less attributes than classical reducts. On the other hand, they may generate if-then rules that make mistakes even within the training samples. For noisy datasets it is to some extent desirable. Nevertheless, some methods for controlling those mistakes should be considered. For example, if the goal is to construct a classification model based on several approximate decision reducts, then – by following ideas taken from machine learning [13] – one may wish to assure that if-then rules generated by different reducts do not repeat the same mistakes on the training data. For this purpose, we can consider a mechanism aiming at diversification of importance of particular objects while searching for different approximate reducts.

The same mechanisms are used in classifier ensemble methods. These methods perform usually better than their components used independently [2, 12]. Combining classifiers is efficient especially if they are substantially different from each other. One of the approaches is to construct many different classifiers based on possibly least overlapping subsets, which goes along perfectly with the above observations on the rough set based feature subset selection methods [24]. In fact, the feature subsets applied in ensembles can be relatively smaller than in case of a single feature subset approach, if we can guarantee that combination of less accurate classifier components (further referred as weak classifiers) will lead back to satisfactory level of determining decision or preserving information about decision. It is important because the outcomes of even most complicated non-symbolic data mining methods become better understandable if computed over smaller subsets of features. Another approach is to create a family of weak classifiers where each consecutive classifier is trained on data where importance of previously unclassified objects was increased. This way one can guarantee that each weak classifier is trained to recognize specific and different objects. This way each weak classifier is diverse from other members of the ensemble.

In our previous work [28], we discussed computing approximate reducts using datasets where objects were labeled with non-negative weights. We showed that attribute subset quality functions used during the process of reduct derivation can be employed in the same way for weighted and non-

weighted objects. In this article we show how this idea can be incorporated into classifier ensemble construction algorithms. We base our experiments on well known Adaptive Boosting (AdaBoost) algorithm [8, 7]. We introduce an AdaBoost version which use decision rules derived from approximate decision reducts. We use voting mechanisms described in [23]. Introduced version can be used for multi-class data. We analyze two different approaches for approximate reduct calculation. The first based on user preference related to reduct size and the second based on approximation threshold. Our experiments are based on benchmark data.

2 APPROXIMATE REDUCTS

Let us introduce basic framework for representing qualitative datasets. By a decision system we mean a tuple $\mathbb{A} = (U, A \cup \{d\})$, where U is a finite set of objects, A is a finite set of attributes and $d \notin A$ is a distinguished decision attribute. We refer to elements of U using their ordinal numbers $i = 1, \dots, |U|$. We treat attributes $a \in A$ as functions $a : U \rightarrow V_a$, where V_a denotes the a 's domain. Values $v_d \in V_d$ correspond to decision classes that we want to describe using the values of attributes in A .

Each subset of attributes $B \subseteq A$ yields a set of decision rules based on all combinations of its values in U . Rules' left sides correspond to equivalence classes $E \in U/IND(B)$ where $IND(B) = \{(i, j) \in U \times U : \forall_{a \in B} a(i) = a(j)\}$ is so called indiscernibility relation. For simplicity, we will write U/B instead of $U/IND(B)$. Rules' right sides can be defined using various methods, e.g., as the following decision classes $X_E \in U/\{d\}$ occurring most often within particular blocks $E \in U/B$:

$$X_E = \operatorname{argmax}_{X \in U/\{d\}} |X \cap E| \quad (1)$$

The elements of $U/\{d\}$ correspond to the values of d . The ratio of objects in U , which are correctly classified by if-then rules produced by B , takes the following form:

$$M(B) = \frac{1}{|U|} \sum_{E \in U/B} |X_E \cap E| \quad (2)$$

We say that $B \subseteq A$ is a decision reduct for \mathbb{A} , iff it is an irreducible subset of attributes such that each pair $i, j \in U$ satisfying inequality $d(i) \neq d(j)$ is discerned by B , i.e., i and j belong to different elements of U/B . In inconsistent decision systems, where even the whole A is not enough to

determine decisions, the constraint for B can be modified, e.g., subject to only those pairs $i, j \in U, d(i) \neq d(j)$, that can be discerned by A . Furthermore, one can consider irreducible subsets that discern *almost all* pairs of objects from different decision classes. In general, there is a great variety of criteria that can be followed while searching for meaningful subsets of attributes.

For a classical decision reduct $B \subseteq A$ in a consistent decision system, there is always equality $M(B) = 1$. However, it is not the case once we allow rules generated by B to be not necessarily deterministic. In [21], it was proposed to use M explicitly for the formulation of the conditions for approximate attribute reduction:

Definition 1. [21] Let $\varepsilon \in [0, 1)$ and $\mathbb{A} = (U, A \cup \{d\})$ be given. We say that $B \subseteq A$ is an (M, ε) -approximate decision reduct, iff it is an irreducible subset of attributes satisfying the following condition:

$$M(B) \geq (1 - \varepsilon)M(A) \quad (3)$$

The problem of finding the smallest (M, ε) -approximate decision reducts is NP-hard, for each $\varepsilon \in [0, 1)$ treated as a constant in the problem's specification [21]. On the other hand, M is monotonic, i.e. $M(C) \leq M(B)$ for any $C \subseteq B$, which makes it possible to adapt some classical reduct search heuristics discussed, e.g., in [1]. Surely, M is not the only possible attribute subset quality function that could be employed in the above inequality. Nevertheless, in the rest of this paper, we concentrate on function M as one of the most natural and generic choices, showing that it can be actually utilized to model quite many strategies of attribute selection and classifier construction.

2.1 Object Weighted Reducts

Attribute subset quality functions provide good basis for evaluating degrees of determining decision classes. However, the original approximate reduct criteria do not allow for controlling which parts of data are problematic for particular reducts. They do not provide the means for expressing importance of objects in a dataset either. Let us propose a general mechanism for expressing objects' importance, based on an arbitrary weight function $\omega : U \rightarrow [0, +\infty)$. Let us reformulate the notion of cardinality of a subset of objects $Y \subseteq U$ according to the following simple definition:

$$|Y|_\omega = \sum_{u \in Y} \omega(u) \quad (4)$$

We rephrase the formula for function M from the previous section as follows, with implicit assumption that there is at least one $u \in U$ such that $\omega(u) > 0$:

$$M_\omega(B) = \frac{1}{|U|_\omega} \sum_{E \in U/B} |X_E^\omega \cap E|_\omega \quad (5)$$

where

$$X_E^\omega = \operatorname{argmax}_{X \in U/\{d\}} |X \cap E|_\omega \quad (6)$$

For a trivial constant function denoted as $\mathbf{1} : U \rightarrow \{1\}$ we have $M_1(B) = M(B)$. Also, for a classical decision reduct $B \subseteq A$ in a consistent decision system there is always $M_\omega(B) = 1$. More specific characteristics can be formulated for the cases of $\omega : U \rightarrow [0, +\infty)$ and $\omega : U \rightarrow (0, +\infty)$. Letting $\omega(u) = 0$ for some objects $u \in U$ may be actually compared to operations on bireducts considered in [25].

Let us formulate several basic properties of M_ω . We start by noting that, for a given $E \in U/B$, there may be several decision classes satisfying (6). Thus, we propose to consider function $\partial_\omega : U/B \rightarrow 2^{U/\{d\}}$ defined as follows, for $E \in U/B$:

$$\partial_\omega(E) = \{X \in U/\{d\} : \forall X' \in U/\{d\} |X \cap E|_\omega \geq |X' \cap E|_\omega\} \quad (7)$$

Definition 2. Let $\varepsilon \in [0, 1)$, $\mathbb{A} = (U, A \cup \{d\})$ and $\omega : U \rightarrow [0, +\infty)$ be given. We say that $B \subseteq A$ is an (ω, ε) -approximate decision reduct, iff it is an irreducible subset of attributes satisfying the following condition:

$$M_\omega(B) \geq (1 - \varepsilon)M_\omega(A) \quad (8)$$

Let us once again denote that the problem of finding an (ω, ε) -approximate decision reduct with minimum number of attributes for an input decision system $\mathbb{A} = (U, A \cup \{d\})$ and $\omega : U \rightarrow [0, +\infty)$ is NP-hard. For fast calculation of approximate reducts we need to use heuristic methods. Algorithm 1 presents a variation of *REDORD* algorithm a bit modified compared to its original form [22].

There can be an exponential number of (ω, ε) -reducts for a given \mathbb{A} and ω . Moreover, different object weights can yield different (ω, ε) -reducts for the same data. Modern rough set approaches to knowledge representation and classifier construction are usually based on ensembles of heuristically found reducts and the corresponding if-then rules with coefficients calculated from the training data. Those coefficients are then used while voting

Algorithm 1 Permutation-based (M_ω, ε) -REDORD (a bit modified comparing to [22])

Input: $\varepsilon \in [0, 1)$; $\mathbb{A} = (U, A \cup \{d\})$; $\sigma : \{1, \dots, n\} \rightarrow \{1, \dots, n\}$; $n = |A|$

Output: $B \subseteq A$

```

1:  $B \leftarrow A$ 
2: for  $i = 1 \rightarrow n$  do
3:   if  $M_\omega(B \setminus \{a_{\sigma(i)}\}) \geq (1 - \varepsilon)M_\omega(A) + \varepsilon M_\omega(\emptyset)$  then
4:      $B \leftarrow B \setminus \{a_{\sigma(i)}\}$ 
5:   end if
6: end for
7: return  $B$ 

```

Table 1. Six options of weighting decisions by if-then rules, corresponding to the consequent coefficient types plain, ω -confidence and ω -coverage, and antecedent coefficient types single and ω -support. $|E|_\omega$ denotes the support of a rule's left side. X_E^ω is defined by formula (6).

| | single | ω -support |
|----------------------|--|--|
| plain | 1 | $ E _\omega / U _\omega$ |
| ω -confidence | $ X_E^\omega \cap E _\omega / E _\omega$ | $ X_E^\omega \cap E _\omega / U _\omega$ |
| ω -coverage | $(X_E^\omega \cap E _\omega / X_E^\omega _\omega) / (E _\omega / U _\omega)$ | $ X_E^\omega \cap E _\omega / X_E^\omega _\omega$ |

about new objects. – A new object is assigned to a decision class with the highest sum of coefficients produced by rules which match its values.

Table 1 illustrates six examples of coefficients that can be assigned by an if-then rule with its left side supported by indiscernibility class $E \in U/B$ to a decision class $X_E^\omega \in U/\{d\}$ identified by formula (6) and used in our previous research [23]. They are analogous to some of other voting strategies already discussed in the rough set literature [1]. Precisely, we consider three possibilities to assign a voting degree to a given decision class: plain, ω -confidence and ω -coverage. We can additionally multiply the rule's vote by its normalized left side's support $|E|_\omega / |U|_\omega$ (antecedent voting type ω -support) or not (single). In our experiments each weak classifier was fixed with single ω -confidence voting method to point to a single output decision.

2.2 Generalized decision and approximate reducts

Let us formulate several basic properties of M_ω . We start by noting that, for a given $E \in U/B$, there may be several decision classes satisfying (6). Thus, we propose to consider function $\partial_\omega : U/B \rightarrow 2^{U/\{d\}}$ defined as follows, for $E \in U/B$:

$$\partial_\omega(E) = \{X \in U/\{d\} : \forall_{X' \in U/\{d\}} |X \cap E|_\omega \geq |X' \cap E|_\omega\} \quad (9)$$

Let us note that ∂_ω is a simple modification of so called generalized decision function introduced within the classical rough set framework [18]. For

a given $B \subseteq A$ and $E \in U/B$, the choice of specific $X_E^\omega \in \partial_\omega(E)$ does not influence the quantity of $M_\omega(B)$. However, function ∂_ω analogously to the original rough set discernibility criteria is also monotonic, i.e. $M_\omega(B) \geq M_\omega(C)$, for $B, C \subseteq A$ such that $C \subseteq B$ and $\omega : U \rightarrow [0, +\infty)$. Moreover, the equality $M_\omega(B) = M_\omega(C)$ takes place, iff for each $E \in U/C$ the intersection of all sets $\partial_\omega(E')$ such that $E' \in U/B$ and $E' \subseteq E$ is not empty. Algorithm 2 explains how one can compute approximate reducts based on concept of generalized decisions. From the user perspective it is different from the method described in the previous section. We do not need to specify the approximation threshold ε . Instead, we specify the maximum size m of a reduct, this might be more suitable when during model construction we need to consider number of attributes used in the model.

Algorithm 2 Permutation based algorithm for calculating approximate reducts with concept of generalized decisions

Input: $\mathbb{A} = (U, A \cup \{d\}), \sigma : \{1, \dots, n\} \rightarrow \{1, \dots, n\}, m \leq |A|$

Output: $B \subseteq A$

```

1:  $B \leftarrow \emptyset$ 
2: for  $i = 1 \rightarrow m$  do
3:    $B \leftarrow \{a_{\sigma(i)}\}$ 
4: end for
5: for  $i = 1 \rightarrow m$  do
6:   if CHECKREMOVEATTRIBUTE( $B, a_{\sigma(i)}$ ) = true then
7:      $B \leftarrow B \setminus \{a_{\sigma(i)}\}$ 
8:   end if
9: end for
10: return  $B$ 

11: function CHECKREMOVEATTRIBUTE( $B, a$ )
12:   for all  $E \in U/B \setminus \{a\}$  do
13:      $DEC \leftarrow V_d$ 
14:     for all  $E' \in U/B, E' \subset E$  do
15:        $DEC \leftarrow DEC \cap \operatorname{argmax}_{X \in U/\{d\}} |X \cap E'|$ 
16:     end for
17:     if  $DEC = \emptyset$  then
18:       return 0
19:     end if
20:   end for
21:   return 1
22: end function

```

3 CLASSIFIER ENSEMBLES

In machine learning, the term *ensemble* refers to multiple instances of algorithms that work together to improve the decision model's performance [3, 19]. Outputs of each of algorithms in the ensemble, further referred as *base* or *weak* classifiers, are combined using various techniques into one common result. The rationale for such approach is that it is more difficult to optimize the design of a general model than in case of combination of relatively simpler classifiers focusing on particular aspects of data. Using the classifier ensembles as decision models offers a simple yet efficient technique for obtaining increased levels of final accuracy. Figure 1 illustrates a high-level architecture of the ensemble decision model.

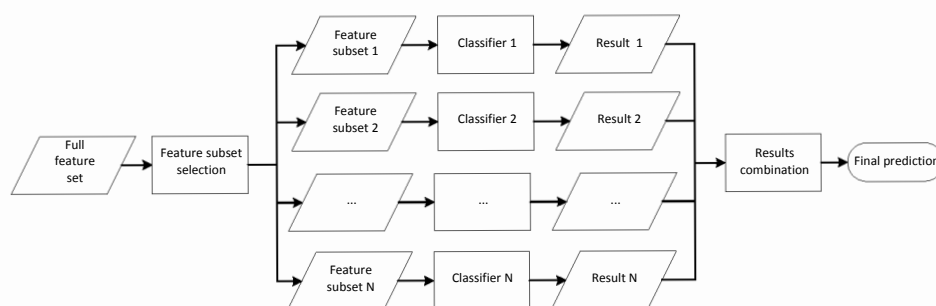


Fig. 1. Prediction model based on the ensemble of feature subsets.

Ensemble accuracy depends both on the quality of the problem decomposition and individual accuracies of base classifiers. Decomposition refers to the way base classifiers are trained. Individual accuracies are very important but, as shown in [14, 11], in order to improve final accuracy there must exist some form of diversity among base classifiers. Indeed, if all base classifiers behave in nearly the same way, little is achieved by combining their mechanisms.

In machine learning, diversity is usually considered by means of classifiers' *output*. Classifiers are regarded as different from output perspective, if they address different, mutually complementary classification aspects or subspaces of instances. A popular method based on diversity of classification aspects is to decompose a multi-class classification problem into a set of two-class sub-problems, each of them handled by a different base classifier [4]. Another example relates to a usage of domain knowledge to decompose the nature of decisions into a hierarchy of layers that can be

addressed more easily [17]. On the other hand, among a number of rough set inspired methods based on the output diversity understood by means of subspaces of instances, there are ensembles of decision bireducts – families of irreducible feature subsets determining decisions subject to minimally overlapping sets of outliers [25].

Diversity can be expressed also in terms of the algorithms' *design* and classifiers' *input* [12]. Both aspects can strengthen the above-discussed output complementarity, although they may have different implications for interactions with humans. The design aspect of diversity relates to the types of algorithms used to train base classifiers. Although it may highly improve the overall model's accuracy, the model's clarity may decrease, making interactions harder. The case of the input aspect of diversity is quite opposite. From this perspective, classifiers are considered as different when they produce their predictions based on different feature subsets or different instance subspaces.

Out of a number of methodologies of classifier ensemble construction, let us focus on boosting [6] and bagging [2]. Both algorithms are based on manipulating the training set samples. Boosting method works by repeatedly running a learning algorithm on variously distributed training data. In each iteration, the number of instances misclassified by a previously produced classifier is used to prepare the training set for the next iteration in such a way that misclassified cases are prioritized. The classifier constructed in the next iteration is focused on instances with higher priority value. In case of bagging (bootstrap aggregating), the datasets created from the original training set may be mutually disjoint or overlapping. The base classifiers are trained over the sample of instances. Classifiers are built independently, while in boosting each classifier is influenced by the performance of those that were built prior to its construction.

Boosting has been a very successful technique for solving classification problems. We present the AdaBoost algorithm in listing 3. AdaBoost was originally designed to solve two-class classification problems but since its first publication many algorithm modifications have been made, mainly restricted to reducing the multi-class classification to multiple two-class problems [9, 20]. There exist however AdaBoost based algorithms that are able to deal with multi-class classification problems directly. In [30] an algorithm called *SAMME* was discussed. We follow the same suggestions related to calculation of weak classifier confidence α (line 8) and setting the error level threshold (line 5). Let us draw reader attention to the fact we set the error threshold $\epsilon < (1 - M_\omega(\emptyset))$ where $M_\omega(\emptyset)$ is the quality

measure discussed in previous section calculated for an empty reduct. The original version of AdaBoost use $\epsilon_t = 1/2$. By using this threshold we make sure that each weak classifier performs better than a classifier predicting decision class only considering a priori decision class distribution.

Algorithm 3 Algorithm AdaBoost

Input: sequence of n examples $\langle (x_1, y_1), \dots, (x_n, y_n) \rangle$ with labels $y_i \in Y = \{1, \dots, k\}$; weak learning algorithm **WeakLearn**; integer T specifying number of iterations;

Output: the final hypothesis $h_{fin}(x) = \arg \max_{y \in Y} \sum_{t: h_t(x)=y} \alpha_t$

- 1: **Initialize:** $\omega_i = 1/n$ for $i = 1, 2, 3, \dots, n$
 - 2: **for** $t = 1 \rightarrow T$ **do**
 - 3: Call **WeakLearn**, providing it with object weights ω_i
 - 4: Get back a hypothesis $h_t : X \rightarrow Y$
 - 5: Calculate the error of h_t : $\epsilon_t \leftarrow \sum_{i: h_t(x_i) \neq y_i} \omega_i$
 - 6: **if** $\epsilon_t > (1 - M_\omega(\emptyset))$ **or** $\epsilon_t = 0$ **then**
 - 7: Break
 - 8: **end if**
 - 9: Set $\alpha_t \leftarrow \ln(\epsilon_t / (1 - \epsilon_t)) + \ln(k - 1)$
 - 10: Update weights: $\omega_i \leftarrow \omega_i \times \begin{cases} \exp(-\alpha_t) & h_t(x_i) = y_i \\ \exp(\alpha_t) & h_t(x_i) \neq y_i \end{cases}$
 - 11: Normalize ω_i
 - 12: **end for**
 - 13: Normalize α
-

4 ENSEMBLES OF APPROXIMATE REDUCTS

Let us now explain how rough sets can be used in classifier ensemble construction. First of all, rough set techniques allow for producing many feature subsets in form of decision reducts, which determine predefined decision classes according to some specified criteria. This assures the quality of feature subsets used as inputs in further learning process. Strongly correlated features are then expected to occur exchangeably in the discovered decision reducts. The same happens for subsets of features that enable to reason about decisions from different angles. This results in the ensembles consisting of complementary yet simple feature subset based classifiers. Discovered feature subsets can later be used as an input to different classifier methods or directly projected onto training data to produce decision rules. We follow the later approach, mainly we calculate approximate decision reducts in an iterative manner. Each reduct is then used to produce decision rules which form a single classifier, member of the ensemble.

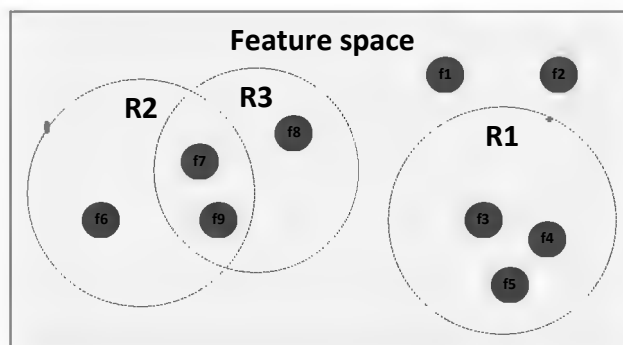


Fig. 2. A dataset with nine features and exactly three decision reducts: R1, R2, and R3. For permutation-based algorithm proposed in [22, 24], almost 43% out of $11!$ feature permutations result with R1, while slightly over 28.5% of permutations result with each of R2 and R3.

Certainly, as already mentioned in Section 2, extraction of the most interesting decision reducts from data is not easy. There are a number of theoretical results emphasizing the complexity of search for minimal (optimal) or generating all (almost all) decision reducts (see e.g. [16, 18]). We should also remember that it is highly desirable to create easy interpretable and easy understandable decision or classification models. In [22, 24], in order to search for decision reducts ensembles, we used a heuristic algorithm based on randomly generated permutations of features. For each tested permutation, the procedure analogous to well-known backward feature elimination (see e.g. [10]) was applied to construct the corresponding decision reduct. Features were tested in the reversed order of appearance in permutation, with (approximate) decision determination criteria examined every time. Algorithm 4 describes in detail our method.

Figure 2 shows a potential of the above technique to promote ensembles with high input diversity with respect to the subsets of features. It illustrates a dataset with only three existing decision reducts (R1, R2, R3). As R1 and R2 are overlapping, it should be preferred to include either R1 and R3, or R2 and R3 into the ensemble. The discovery process can be optimized by intelligent search operators (see e.g. [29]) or adding optimization criteria reflecting other types of ensemble diversity (see e.g. [23]).

Feature permutation based methodology can be also generalized onto discovery of other forms of data based knowledge representation, such as approximate functional dependencies [22] or already mentioned decision bireducts [25] that can be, actually, treated as a rough set inspired adaptation of the ideas of bagging and boosting. Another important aspect of

Algorithm 4 AdaBoost with (ω, ε) -Approximate Reducts as Weak Classifier

Input: $\mathbb{A} = (U, A \cup \{d\})$, $n = |A|$, $\varepsilon \in [0, 1)$
integer T specifying number of iterations
Output: Approximate Reduct Ensemble $S = \{r_1, \dots, r_s\}$, $s \leq T$
Initialize: $\omega_i = 1/n$ for $i = 1, 2, 3, \dots, n$

- 1: Calculate error threshold $\epsilon_0 = 1 - M_\omega(\emptyset)$;
- 2: **for** $t = 1 \rightarrow T$ **do**
- 3: Generate permutation σ
- 4: Create Approximate Reducts r_t based on permutation σ
- 5: Generate decision rules based on conditional attributes from reduct r_t
- 6: Classify training examples using voting mechanisms identifying training labels
- 7: Calculate the error ϵ_t
- 8: **if** $\epsilon_t > \epsilon_0$ **or** $\epsilon_t = 0$ **then**
- 9: Break
- 10: **end if**
- 11: Calculate weak classifier confidence α_t
- 12: Update and normalize object weights ω
- 13: **end for**
- 14: Normalize α

using a generic mechanism based on permutations is its ability to take into account domain expert preferences. Let us note that features occurring in front of a permutation are more likely to get included into the resulting decision reduct. We use this fact and try to construct permutations in such a way that attributes already existing in the model (occurring in reducts calculated in previous iterations) are less likely to be selected at the beginning of the permutations.

5 EXPERIMENTS

We conducted our experiments on a collection of benchmark datasets available from the University of California at Irvine (UCI) Repository [15]. The summary of datasets is given in Table 2. Some datasets were already partitioned into train and test sets, for others we used 5-fold cross validation. We also included data with missing values. For the training sets the missing values were replaced with the most frequent value within a particular class label. For the testing set the missing values were not replaced and were treated as unknown value. This way when an object with an attribute with missing value needs to be classified, only rules not including this particular attribute could be applied. All results have been averaged over at least 20 runs. In case of cross validation procedure that means at least 100 runs for each algorithm on each dataset. We tested each algorithm against the number of iterations i where $i = 1, 2, 5, 10, 20, 50, 100$.

Table 2. UCI Benchmark datasets used in the experiments.

| dataset name | # examples | | # classes | # attributes | | class distribution train (test) | missing values |
|------------------|------------|------|-----------|--------------|-------|--|----------------|
| | train | test | | disc. | cont. | | |
| soybean-small | 47 | - | 4 | 35 | - | 10:10:10:17 | - |
| heart-spect | 80 | 187 | 2 | 22 | - | 54:26 (103:84) | - |
| zoo | 101 | - | 7 | 17 | - | 41:20:5:13:4:8:10 | - |
| promoters | 106 | - | 2 | 57 | - | 53:53 | - |
| monks-3 | 122 | 432 | 2 | 6 | - | 62:60 (204:228) | - |
| monks-1 | 123 | 432 | 2 | 6 | - | 62:62 (216:216) | - |
| monks-2 | 169 | 432 | 2 | 6 | - | 105:64 (289:142) | - |
| audiology.stand | 200 | 26 | 24 | 69 | - | 1:1:46:18:1:2:16:48:2:6:1:4:2:2:5:3:1:2:20:4:1:4:8:2 (0:0:11:4:0:0:4:0:0:0:1:0:0:0:4:0:0:0:2:0:0:0:0) | ✓ |
| soybean-large | 307 | 376 | 19 | 35 | - | 10:10:10:40:20:10:10:40:10:10:10:20:10:40:40:6:6:1:4 (10:10:10:48:24:10:10:52:10:10:10:24:10:51:51:9:8:15:4) | ✓ |
| house-votes-84 | 435 | - | 2 | 16 | - | 267:168 | ✓ |
| breast-cancer-w | 699 | - | 2 | - | 9 | 458:241 | ✓ |
| semeion | 1593 | - | 10 | 257 | - | 161:162:159:159:161:159:161:158:155:158 | - |
| dna (splice) | 2000 | 1186 | 3 | 20 | - | 464:485:1051 (303:280:603) | - |
| kr-vs-kp (chess) | 3196 | - | 2 | 36 | - | 1527:1669 | - |
| optdigit-recogn | 3823 | 1797 | 10 | - | 64 | 376:389:380:389:387:376:377:387:380:382 (178:182:177:183:181:182:181:179:174:180) | - |
| pendigit-recogn | 7494 | 3498 | 10 | - | 16 | 780:779:780:719:780:720:720:778:719:719 (363:364:364:336:364:335:336:364:336:336) | - |
| agaricus-lepiot | 8124 | - | 2 | 22 | - | 4208:3916 | - |
| nursery | 12960 | - | 5 | 8 | - | 4320:2:328:4266:4044 | - |
| letter-recogn | 15000 | 5000 | 26 | - | 16 | 594:567:554:598:565:565:547:538:567:575:572: 560:607:605:574:597:594:568:563:599:606:581:576:583:602:543 (195:199:182:207:203:210:226:196:188:172:167:201:185: 178:179:206:189:190:185:197:207:183:176:204:184:191) | - |

The first bunch of experiments we conducted was to check the results obtained for approximate reducts derived using described generalized decision approach, referenced as Approximate Reduct Ensemble (*ARE*). We fixed the initial permutation size to $\lfloor \log_2(m + 1) \rfloor$, where m is the total number of conditional attributes. We set such small number of attributes in order to be able to compare this results with the other described methods utilizing approximation threshold. The thresholds used in later experiments led to comparable reduct sizes. However, the reader should be aware that usually weak classifiers combined into ensembles use more attributes when calculating classification models. The results of the first experiment bunch are presented in Table 3. Table presents error rates obtained for different datasets and number of iterations. Additionally, we tested a procedure for diversifying attributes when constructing attribute permutations which takes into account attributes already selected for classification model, referenced as Approximate Reduct Ensemble with Diversity (*AREwD*). This procedure was described in section 4. For results comparison we include a single classifier error rate encoded as iteration $i = 1$, in which case there is

no difference if *ARE* or *AREwD* method is used. We bold out the result with lower error rate on the test set. The last row sum up number of wins for a particular method.

In general *AREwD* method seems to perform a bit better than *ARE*. The win-tie-lose numbers for *AREwD* were 65-6-43. However both methods have some drawbacks. First of all for datasets like *promoters*, *audiology.stand*, *optdigit-recogn* and *pendigit-recogn* the results were even worst than the blind guess. In each case the results on training set were good which might suggest that the learning algorithm developed a model that was overfitted with training data. Another drawback for both methods is that there is no direct relation between attribute selection and the obtained classification error. In each iteration the attributes are selected randomly based on drawn permutation. There is no mechanism that would allow to select meaningful attributes over the redundant and unimportant ones except the step of reduct reduction which in this case happens rather rarely.

The second bunch of experiments we conducted was to test AdaBoost with the second method of deriving approximate reducts introduced in this paper. Table 4 shows the results of experiments with (ε, ω) -approximate decision reducts calculated over universe of weighted objects, referenced as epsilon-Approximate Reduct Ensemble (*eARE*). Similarly to the previous experiment we also evaluate two types of attribute permutation construction, random permutation selection and selection that favours attributes that are not yet selected in the classification model. This second type is encoded as *eAREwD*. We calculated ε using the following formula: $\varepsilon_{0.5} = (1 - M(\emptyset))/2$. Compared to *ARE* and *AREwD* the results were improved for several datasets. In few cases error rates were increased which is due to the fact that the approximation threshold was too high. Similarly to the previous bunch of experiments *eAREwD* performs slightly better. The win-tie-lose numbers for *eAREwD* were 82-10-22.

In the last bunch of experiments we set the approximation threshold to $\varepsilon_K = (1 - M(\emptyset))/k$, where k is the number of decision classes. Table 5 presents the obtained error rates. The datasets with only two decision classes were eliminated from the table as the results for these cases are already presented in Table 4. The win-tie-lose numbers for *eAREwD* method were 36-3-21. The obtained results compared to the second experiment bunch were improved in few cases.

Table 6 presents average reduct length in ensemble obtained for *eARE* and *eAREwD* methods for approximation thresholds $\varepsilon_{0.5}$ and ε_k . In both

Table 3. Error rates obtained for ARE and AREwD methods.

| # iterations method dataset | 1 | | 2 | | | | 5 | | | | 10 | | | | | | | | | | | | | | | |
|-----------------------------------|-------|------|------|-------------|------|--------------|------|-------------|------|-------------|------|-------------|------|-------------|----|--|--|--|----|--|--|--|----|--|--|--|
| | train | test | ARE | AREwD | ARE | AREwD | ARE | AREwD | ARE | AREwD | ARE | AREwD | ARE | AREwD | | | | | | | | | | | | |
| soybean-small | 20.7 | 32.2 | 21.9 | 34.1 | 18.9 | 26.7 | 4.4 | 15.5 | 7.8 | 18.3 | 1.5 | 7.8 | 1.6 | 8.6 | | | | | | | | | | | | |
| heart-spect | 24.5 | 41.0 | 44.9 | 47.9 | 38.3 | 44.6 | 21.8 | 41.2 | 21.7 | 38.5 | 34.9 | 43.0 | 28.5 | 41.7 | | | | | | | | | | | | |
| zoo | 20.1 | 25.1 | 24.2 | 30.6 | 26.1 | 31.0 | 11.6 | 18.5 | 9.8 | 17.5 | 2.8 | 8.8 | 2.4 | 8.9 | | | | | | | | | | | | |
| promoters | 1.8 | 90.5 | 1.2 | 88.7 | 1.2 | 91.2 | 0.0 | 91.8 | 0.0 | 93.2 | 0.0 | 91.7 | 0.0 | 93.9 | | | | | | | | | | | | |
| monks-3 | 29.9 | 30.0 | 40.5 | 42.9 | 31.8 | 34.2 | 17.3 | 18.1 | 17.8 | 17.8 | 15.6 | 17.5 | 15.5 | 18.9 | | | | | | | | | | | | |
| monks-1 | 32.8 | 39.5 | 35.4 | 41.0 | 33.6 | 39.4 | 30.5 | 34.5 | 28.3 | 33.9 | 27.2 | 29.0 | 26.1 | 30.9 | | | | | | | | | | | | |
| monks-2 | 36.8 | 35.3 | 43.9 | 47.4 | 47.1 | 51.0 | 36.7 | 38.8 | 37.2 | 40.1 | 38.8 | 41.1 | 39.2 | 41.3 | | | | | | | | | | | | |
| audiology.stand | 64.7 | 74.4 | 63.4 | 68.3 | 65.8 | 78.3 | 63.8 | 76.3 | 59.1 | 71.0 | 56.7 | 68.1 | 52.0 | 66.2 | | | | | | | | | | | | |
| soybean-large | 47.4 | 64.6 | 47.9 | 63.4 | 40.9 | 59.6 | 31.5 | 58.0 | 30.3 | 56.4 | 13.2 | 45.7 | 12.1 | 44.3 | | | | | | | | | | | | |
| house-votes-84 | 9.5 | 23.0 | 16.7 | 28.0 | 13.1 | 26.8 | 8.7 | 25.6 | 7.3 | 24.1 | 5.7 | 23.3 | 5.1 | 21.5 | | | | | | | | | | | | |
| breast-cancer-w | 2.0 | 23.7 | 7.8 | 28.8 | 4.5 | 25.8 | 0.9 | 23.3 | 0.3 | 22.7 | 0.0 | 21.8 | 0.0 | 22.1 | | | | | | | | | | | | |
| semeion | 48.5 | 59.7 | 48.8 | 60.2 | 49.3 | 60.8 | 29.4 | 49.3 | 28.3 | 48.0 | 14.7 | 40.7 | 13.7 | 39.8 | | | | | | | | | | | | |
| dna (splice) | 26.5 | 32.0 | 31.3 | 35.7 | 27.7 | 31.9 | 17.1 | 26.8 | 13.9 | 23.5 | 7.9 | 20.4 | 6.2 | 18.4 | | | | | | | | | | | | |
| kr-vs-kp (chess) | 36.9 | 37.7 | 36.8 | 37.3 | 35.8 | 36.1 | 28.8 | 29.7 | 27.6 | 28.3 | 23.0 | 23.7 | 22.6 | 23.4 | | | | | | | | | | | | |
| optdigit-recogn | 10.9 | 84.5 | 14.0 | 81.2 | 9.2 | 85.7 | 0.9 | 87.4 | 0.9 | 87.9 | 0.0 | 90.0 | 0.0 | 91.9 | | | | | | | | | | | | |
| pendigit-recogn | 0.7 | 95.5 | 0.8 | 94.8 | 1.1 | 94.8 | 0.0 | 99.4 | 0.0 | 99.7 | 0.0 | 99.9 | 0.0 | 100.0 | | | | | | | | | | | | |
| agaricus-lepiot | 9.1 | 12.4 | 12.6 | 17.3 | 13.2 | 16.9 | 3.5 | 5.9 | 4.0 | 6.1 | 0.8 | 2.5 | 0.9 | 1.1 | | | | | | | | | | | | |
| nursery | 43.0 | 43.5 | 43.8 | 44.0 | 42.1 | 42.3 | 32.7 | 32.8 | 31.5 | 32.0 | 25.9 | 26.2 | 25.9 | 26.5 | | | | | | | | | | | | |
| letter-recogn | 40.6 | 58.2 | 34.5 | 51.8 | 35.5 | 52.3 | 13.1 | 41.0 | 12.4 | 41.3 | 3.0 | 31.8 | 2.8 | 32.2 | | | | | | | | | | | | |
| # wins | - | | 7 | | | | 11 | | | | 7 | | | | 12 | | | | 11 | | | | 8 | | | |
| # iterations | 1 | | 20 | | | | 50 | | | | 100 | | | | | | | | | | | | | | | |
| soybean-small | 20.7 | 32.2 | 0.0 | 4.2 | 0.2 | 4.0 | 0.0 | 5.5 | 0.0 | 3.5 | 0.6 | 7.1 | 0.0 | 4.5 | | | | | | | | | | | | |
| heart-spect | 24.5 | 41.0 | 24.1 | 41.4 | 21.6 | 40.6 | 20.9 | 37.5 | 19.9 | 38.5 | 21.2 | 39.5 | 20.3 | 37.9 | | | | | | | | | | | | |
| zoo | 20.1 | 25.1 | 0.2 | 4.7 | 0.6 | 5.0 | 0.9 | 4.4 | 0.0 | 3.7 | 0.3 | 2.7 | 0.5 | 4.3 | | | | | | | | | | | | |
| promoters | 1.8 | 90.5 | 0.0 | 93.6 | 0.0 | 93.1 | 0.0 | 93.7 | 0.0 | 94.4 | 0.0 | 92.2 | 0.0 | 93.1 | | | | | | | | | | | | |
| monks-3 | 29.9 | 30.0 | 9.0 | 14.2 | 9.9 | 13.7 | 3.5 | 9.7 | 3.5 | 9.2 | 0.3 | 6.7 | 0.6 | 7.1 | | | | | | | | | | | | |
| monks-1 | 32.8 | 39.5 | 17.3 | 20.0 | 16.4 | 18.9 | 0.4 | 1.4 | 0.4 | 1.5 | 0.0 | 0.5 | 0.0 | 0.5 | | | | | | | | | | | | |
| monks-2 | 36.8 | 35.3 | 38.1 | 40.8 | 41.0 | 43.4 | 38.5 | 40.5 | 38.9 | 41.9 | 39.6 | 42.0 | 36.1 | 39.8 | | | | | | | | | | | | |
| audiology.stand | 64.7 | 74.4 | 49.5 | 64.2 | 51.9 | 69.2 | 38.0 | 66.9 | 52.7 | 70.2 | 35.2 | 58.7 | 47.1 | 59.0 | | | | | | | | | | | | |
| soybean-large | 47.4 | 64.6 | 5.9 | 38.4 | 5.6 | 37.8 | 2.1 | 32.0 | 1.8 | 32.4 | 0.9 | 31.3 | 0.7 | 30.9 | | | | | | | | | | | | |
| house-votes-84 | 9.5 | 23.0 | 2.9 | 21.8 | 2.6 | 20.5 | 1.0 | 20.8 | 0.8 | 19.2 | 0.7 | 19.5 | 0.4 | 19.4 | | | | | | | | | | | | |
| breast-cancer-w | 2.0 | 23.7 | 0.0 | 21.2 | 0.0 | 21.4 | 0.0 | 20.5 | 0.0 | 21.5 | 0.0 | 21.1 | 0.0 | 21.5 | | | | | | | | | | | | |
| semeion | 48.5 | 59.7 | 2.7 | 29.7 | 2.3 | 28.5 | 0.0 | 17.4 | 0.0 | 16.8 | 0.0 | 12.2 | 0.0 | 11.5 | | | | | | | | | | | | |
| dna (splice) | 26.5 | 32.0 | 1.6 | 13.8 | 1.5 | 14.1 | 0.0 | 10.0 | 0.0 | 9.5 | 0.0 | 6.9 | 0.0 | 6.6 | | | | | | | | | | | | |
| kr-vs-kp (chess) | 36.9 | 37.7 | 17.0 | 18.1 | 16.3 | 17.6 | 9.5 | 10.8 | 9.4 | 10.8 | 5.3 | 6.8 | 5.2 | 6.7 | | | | | | | | | | | | |
| optdigit-recogn | 10.9 | 84.5 | 0.0 | 95.9 | 0.0 | 94.6 | 0.0 | 97.8 | 0.0 | 93.8 | 0.0 | 98.4 | 0.0 | 92.7 | | | | | | | | | | | | |
| pendigit-recogn | 0.7 | 95.5 | 0.0 | 100.0 | 0.0 | 100.0 | 0.0 | 100.0 | 0.0 | 100.0 | 0.0 | 100.0 | 0.0 | 100.0 | | | | | | | | | | | | |
| agaricus-lepiot | 9.1 | 12.4 | 0.0 | 0.3 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.1 | | | | | | | | | | | | |
| nursery | 43.0 | 43.5 | 19.8 | 20.2 | 18.3 | 18.6 | 10.9 | 11.3 | 10.4 | 10.8 | 7.0 | 7.4 | 6.7 | 7.2 | | | | | | | | | | | | |
| letter-recogn | 40.6 | 58.2 | 0.2 | 24.9 | 0.1 | 23.8 | 0.0 | 19.9 | 0.0 | 19.9 | 0.0 | 18.8 | 0.0 | 18.4 | | | | | | | | | | | | |
| # wins | - | | 5 | | | | 14 | | | | 7 | | | | 8 | | | | 6 | | | | 12 | | | |

Table 4. Error rates obtained for $eARE$ and $eAREwD$ methods and $\varepsilon_{0.5}$

| # iterations method dataset | 1 | | 2 | | | | 5 | | | | 10 | | | |
|-----------------------------------|-------|------|---------------|--------------|-----------------|----------------|---------------|--------------|-----------------|----------------|---------------|--------------|-----------------|----------------|
| | train | test | eARE train | eARE test | eAREwD train | eAREwD test | eARE train | eARE test | eAREwD train | eAREwD test | eARE train | eARE test | eAREwD train | eAREwD test |
| soybean-small | 15.9 | 29.6 | 59.2 | 56.9 | 57.5 | 59.4 | 36.2 | 41.6 | 32.2 | 39.5 | 15.8 | 21.9 | 11.6 | 17.2 |
| heart-spect | 16.2 | 59.6 | 50.9 | 41.9 | 44.1 | 35.2 | 15.4 | 44.8 | 15.6 | 44.1 | 38.8 | 27.9 | 36.4 | 28.5 |
| zoo | 15.1 | 21.3 | 49.4 | 51.5 | 45.7 | 49.4 | 24.3 | 30.9 | 25.1 | 34.9 | 17.0 | 24.6 | 13.7 | 21.7 |
| promoters | 7.4 | 71.4 | 43.9 | 49.2 | 43.9 | 47.3 | 6.1 | 42.5 | 6.0 | 40.9 | 36.0 | 46.1 | 33.0 | 44.0 |
| monks-3 | 7.5 | 19.1 | 55.2 | 60.5 | 62.0 | 67.5 | 7.7 | 25.6 | 7.5 | 19.8 | 26.6 | 32.3 | 24.0 | 28.9 |
| monks-1 | 5.8 | 22.4 | 16.6 | 18.5 | 11.0 | 12.6 | 2.7 | 10.0 | 5.0 | 9.7 | 12.0 | 15.1 | 9.0 | 12.0 |
| monks-2 | 0.0 | 60.9 | 0.0 | 60.9 | 0.0 | 60.9 | 0.0 | 60.9 | 0.0 | 60.9 | 0.0 | 60.9 | 0.0 | 60.9 |
| audiology.stand | 28.5 | 67.5 | 54.5 | 79.6 | 51.2 | 72.1 | 50.2 | 72.9 | 48.2 | 75.2 | 19.5 | 52.9 | 20.5 | 62.5 |
| soybean-large | 36.0 | 58.7 | 39.7 | 62.5 | 45.2 | 64.7 | 46.5 | 65.9 | 46.7 | 65.2 | 36.1 | 57.6 | 35.5 | 59.2 |
| house-votes-84 | 5.7 | 21.1 | 30.5 | 40.6 | 27.7 | 38.5 | 5.4 | 17.7 | 5.6 | 15.9 | 15.1 | 21.9 | 13.2 | 18.7 |
| breast-cancer-w | 4.4 | 10.9 | 17.0 | 25.6 | 11.3 | 20.1 | 2.9 | 8.8 | 3.0 | 8.2 | 6.6 | 10.5 | 5.0 | 9.2 |
| semeion | 38.4 | 58.5 | 44.0 | 62.0 | 43.2 | 61.1 | 33.3 | 51.8 | 33.2 | 51.8 | 28.2 | 47.4 | 28.6 | 47.5 |
| dna (splice) | 9.1 | 59.6 | 17.5 | 51.8 | 17.8 | 50.7 | 7.7 | 30.1 | 8.2 | 31.2 | 6.3 | 24.8 | 5.1 | 20.7 |
| kr-vs-kp (chess) | 11.0 | 25.0 | 27.7 | 32.6 | 23.3 | 27.7 | 8.9 | 16.8 | 8.4 | 15.7 | 15.4 | 18.5 | 14.2 | 17.6 |
| optdigit-recogn | 33.6 | 68.6 | 34.2 | 70.5 | 35.4 | 70.4 | 25.4 | 61.3 | 27.1 | 61.9 | 20.7 | 56.1 | 20.7 | 54.4 |
| pendigit-recogn | 29.5 | 68.6 | 31.1 | 71.4 | 30.9 | 68.7 | 17.4 | 63.6 | 17.1 | 62.9 | 12.0 | 57.0 | 12.4 | 57.0 |
| agaricus-lepiot | 7.8 | 11.2 | 28.2 | 29.6 | 26.5 | 28.9 | 5.9 | 7.0 | 5.3 | 6.5 | 24.5 | 25.1 | 19.1 | 19.3 |
| nursery | 18.6 | 18.9 | 39.9 | 40.0 | 42.1 | 42.1 | 32.0 | 32.5 | 28.8 | 29.6 | 29.6 | 29.9 | 29.5 | 29.9 |
| letter-recogn | 39.2 | 57.3 | 39.8 | 57.9 | 41.0 | 61.2 | 19.8 | 42.9 | 18.4 | 41.8 | 20.1 | 41.5 | 19.4 | 41.2 |
| # wins | | | | 5 | | 13 | | 4 | | 13 | | 4 | | 12 |
| # iterations | | 1 | | 20 | | | | 50 | | | | 100 | | |
| soybean-small | 15.9 | 29.6 | 5.6 | 9.9 | 3.7 | 8.1 | 0.0 | 2.6 | 0.1 | 1.0 | 0.0 | 0.3 | 0.0 | 0.1 |
| heart-spect | 16.2 | 59.6 | 32.6 | 27.5 | 36.4 | 27.3 | 39.9 | 29.3 | 40.2 | 27.1 | 35.6 | 29.3 | 35.1 | 27.7 |
| zoo | 15.1 | 21.3 | 5.3 | 14.3 | 3.6 | 11.2 | 0.9 | 8.6 | 1.1 | 8.4 | 0.1 | 5.8 | 0.1 | 6.5 |
| promoters | 7.4 | 71.4 | 36.0 | 45.4 | 33.5 | 45.1 | 38.0 | 47.4 | 34.2 | 44.7 | 36.2 | 47.1 | 36.8 | 46.3 |
| monks-3 | 7.5 | 19.1 | 19.6 | 27.6 | 30.7 | 36.2 | 20.7 | 27.2 | 17.3 | 23.0 | 29.2 | 35.2 | 19.6 | 25.7 |
| monks-1 | 5.8 | 22.4 | 12.3 | 14.2 | 9.2 | 10.9 | 5.8 | 7.8 | 14.4 | 15.7 | 8.6 | 12.0 | 11.8 | 13.2 |
| monks-2 | 0.0 | 60.9 | 0.0 | 60.9 | 0.0 | 60.9 | 0.0 | 60.9 | 0.0 | 60.9 | 0.0 | 60.9 | 0.0 | 60.9 |
| audiology.stand | 28.5 | 67.5 | 10.4 | 52.9 | 9.9 | 50.8 | 5.6 | 41.9 | 4.3 | 42.9 | 3.0 | 36.5 | 2.9 | 37.3 |
| soybean-large | 36.0 | 58.7 | 20.5 | 49.1 | 21.1 | 49.1 | 11.2 | 41.1 | 10.8 | 40.8 | 7.3 | 36.7 | 7.9 | 37.2 |
| house-votes-84 | 5.7 | 21.1 | 15.7 | 19.5 | 10.5 | 15.6 | 13.6 | 17.4 | 8.0 | 12.0 | 13.8 | 17.9 | 10.0 | 14.0 |
| breast-cancer-w | 4.4 | 10.9 | 6.3 | 10.2 | 4.9 | 8.6 | 5.9 | 9.5 | 4.8 | 8.8 | 6.9 | 11.2 | 5.4 | 9.3 |
| semeion | 38.4 | 58.5 | 20.2 | 40.5 | 20.0 | 40.1 | 9.2 | 31.3 | 9.0 | 30.1 | 3.6 | 23.3 | 3.3 | 22.2 |
| dna (splice) | 9.1 | 59.6 | 3.4 | 18.8 | 3.3 | 16.5 | 1.7 | 12.8 | 1.6 | 12.0 | 0.7 | 9.5 | 0.6 | 9.2 |
| kr-vs-kp (chess) | 11.0 | 25.0 | 12.4 | 16.2 | 12.2 | 15.4 | 12.5 | 15.4 | 10.1 | 12.9 | 12.1 | 15.6 | 10.8 | 14.2 |
| optdigit-recogn | 33.6 | 68.6 | 13.2 | 47.1 | 12.1 | 45.9 | 3.6 | 35.0 | 3.7 | 34.2 | 0.8 | 26.1 | 0.9 | 25.9 |
| pendigit-recogn | 29.5 | 68.6 | 7.3 | 48.3 | 6.7 | 46.1 | 2.9 | 35.1 | 2.7 | 33.7 | 1.2 | 26.9 | 1.2 | 26.6 |
| agaricus-lepiot | 7.8 | 11.2 | 21.6 | 21.6 | 20.5 | 20.3 | 23.4 | 23.7 | 18.6 | 18.9 | 24.6 | 24.0 | 21.0 | 20.9 |
| nursery | 18.6 | 18.9 | 23.0 | 23.5 | 21.8 | 22.6 | 15.3 | 15.6 | 14.9 | 15.4 | 11.4 | 11.8 | 11.7 | 12.1 |
| letter-recogn | 39.2 | 57.3 | 8.6 | 30.9 | 8.1 | 31.1 | 2.8 | 23.1 | 2.9 | 22.7 | 1.2 | 17.7 | 1.2 | 17.5 |
| # wins | | | | 2 | | 15 | | 2 | | 16 | | 5 | | 13 |

Table 5. Error rates obtained for $eARE$ and $eAREwD$ methods and ε_k

| # iterations method dataset | 1 | | 2 | | | | 5 | | | | 10 | | | |
|-----------------------------------|-------|------|---------------|--------------|-----------------|----------------|---------------|--------------|-----------------|----------------|---------------|--------------|-----------------|----------------|
| | train | test | eARE train | eARE test | eAREwD train | eAREwD test | eARE train | eARE test | eAREwD train | eAREwD test | eARE train | eARE test | eAREwD train | eAREwD test |
| soybean-small | 6.2 | 23.3 | 58.7 | 59.1 | 62.1 | 63.7 | 16.0 | 20.5 | 22.5 | 28.3 | 9.4 | 13.4 | 12.5 | 18.4 |
| zoo | 4.1 | 14.8 | 76.2 | 77.9 | 74.4 | 76.1 | 12.3 | 17.8 | 12.5 | 18.5 | 2.9 | 10.4 | 2.6 | 10.4 |
| audiology.stand | 2.0 | 73.1 | 75.3 | 81.9 | 79.5 | 85.8 | 2.3 | 62.1 | 4.0 | 69.2 | 2.0 | 64.6 | 0.8 | 63.3 |
| soybean-large | 3.8 | 72.4 | 58.3 | 76.8 | 55.4 | 75.0 | 5.5 | 65.1 | 7.1 | 62.9 | 3.7 | 58.2 | 4.8 | 55.2 |
| semeion | 7.6 | 74.3 | 24.5 | 66.5 | 25.3 | 66.2 | 7.1 | 58.0 | 7.2 | 59.5 | 1.8 | 51.7 | 1.9 | 51.0 |
| dna (splice) | 4.7 | 70.4 | 9.4 | 66.4 | 11.3 | 66.2 | 8.0 | 40.4 | 8.0 | 35.5 | 4.7 | 25.4 | 4.5 | 23.9 |
| optdigit-recogn | 6.3 | 85.6 | 10.0 | 82.9 | 20.3 | 77.2 | 4.4 | 75.1 | 5.2 | 76.1 | 1.3 | 65.1 | 1.7 | 66.3 |
| pendigit-recogn | 5.0 | 87.0 | 15.4 | 81.5 | 14.4 | 80.5 | 4.8 | 77.7 | 4.4 | 74.9 | 0.7 | 66.8 | 0.9 | 64.3 |
| nursery | 8.0 | 8.5 | 46.9 | 47.8 | 49.1 | 49.6 | 21.3 | 22.3 | 21.7 | 22.6 | 16.7 | 17.5 | 16.8 | 17.8 |
| letter-recogn | 2.8 | 64.1 | 6.0 | 62.4 | 11.7 | 56.5 | 1.5 | 55.3 | 1.8 | 54.6 | 0.2 | 50.6 | 0.2 | 49.8 |
| # wins | | | | 3 | | 7 | | 6 | | 4 | | 3 | | 6 |
| # iterations | 1 | | 20 | | | | 50 | | | | 100 | | | |
| soybean-small | 6.2 | 23.3 | 3.6 | 9.2 | 1.9 | 5.6 | 0.1 | 1.4 | 0.0 | 1.6 | 0.0 | 0.5 | 0.0 | 0.2 |
| zoo | 4.1 | 14.8 | 0.6 | 6.6 | 0.8 | 6.7 | 0.1 | 4.6 | 0.0 | 5.1 | 0.0 | 3.7 | 0.0 | 4.9 |
| audiology.stand | 2.0 | 73.1 | 0.0 | 55.4 | 0.3 | 55.6 | 0.0 | 43.3 | 0.0 | 47.5 | 0.0 | 40.2 | 0.0 | 43.3 |
| soybean-large | 3.8 | 72.4 | 0.7 | 53.3 | 0.6 | 50.9 | 0.1 | 46.6 | 0.1 | 43.6 | 0.1 | 39.6 | 0.1 | 36.3 |
| semeion | 7.6 | 74.3 | 0.4 | 36.9 | 0.4 | 36.1 | 0.0 | 22.5 | 0.0 | 21.1 | 0.0 | 15.3 | 0.0 | 14.1 |
| dna (splice) | 4.7 | 70.4 | 2.3 | 18.6 | 2.3 | 17.2 | 0.9 | 12.9 | 0.7 | 11.5 | 0.3 | 9.8 | 0.3 | 8.8 |
| optdigit-recogn | 6.3 | 85.6 | 0.1 | 52.1 | 0.3 | 50.5 | 0.0 | 31.0 | 0.0 | 28.6 | 0.0 | 21.3 | 0.0 | 20.1 |
| pendigit-recogn | 5.0 | 87.0 | 0.1 | 54.6 | 0.1 | 52.5 | 0.0 | 38.2 | 0.0 | 35.3 | 0.0 | 26.7 | 0.0 | 25.0 |
| nursery | 8.0 | 8.5 | 12.0 | 13.3 | 10.1 | 11.5 | 7.9 | 9.1 | 7.4 | 8.4 | 5.6 | 6.7 | 5.7 | 6.8 |
| letter-recogn | 2.8 | 64.1 | 0.0 | 42.0 | 0.0 | 41.1 | 0.0 | 30.3 | 0.0 | 30.2 | 0.0 | 21.2 | 0.0 | 21.9 |
| # wins | | | | 2 | | 8 | | 3 | | 7 | | 4 | | 6 |

cases the average length decreases when number of iterations increases, which means that with every consecutive iteration we need to provide information based on less attributes. This is somehow coherent with intuition. Having constructed a classification model with a number of attributes in case we need to improve it we do not expect to add more information based on the same number of attributes but we would expect only a portion. In case of ARE and $AREwD$ methods there was no length reduction for reducts selected in the ensemble, no matter what number of iterations was selected.

6 CONCLUSIONS AND FUTURE WORK

Boosting methods combined with approximate reducts calculated on the universe weighted objects can provide tools for constructing efficient clas-

Table 6. Average reduct size in the ensemble for $eARE$ and $eAREwD$ methods

| # iterations threshold method | 1 | | 2 | | | | 5 | | | | 10 | | | |
|-------------------------------------|------------------|--------------|------------------|--------------|--------------|--------------|------------------|-------------|--------------|-------------|------------------|-------------|--------------|-------------|
| | $\epsilon_{0.5}$ | ϵ_k | $\epsilon_{0.5}$ | | ϵ_k | | $\epsilon_{0.5}$ | | ϵ_k | | $\epsilon_{0.5}$ | | ϵ_k | |
| | eARB | eARBwD | eARB | eARBwD | eARB | eARBwD | eARB | eARBwD | eARB | eARBwD | eARB | eARBwD | eARB | eARBwD |
| soybean-small | 2.03 | 2.82 | 1.29 | 1.28 | 1.68 | 1.69 | 0.97 | 1.02 | 1.48 | 1.45 | 0.82 | 0.81 | 1.24 | 1.13 |
| heart-spect | 6.90 | - | 5.43 | 5.68 | - | - | 3.59 | 3.51 | - | - | 2.06 | 2.14 | - | - |
| zoo | 3.10 | 5.22 | 2.88 | 2.77 | 3.11 | 3.10 | 1.88 | 1.94 | 2.77 | 2.89 | 1.59 | 1.65 | 2.19 | 2.28 |
| promoters | 3.75 | - | 2.28 | 2.22 | - | - | 1.34 | 1.27 | - | - | 0.67 | 0.72 | - | - |
| monks-3 | 2.98 | - | 2.10 | 1.78 | - | - | 1.73 | 1.76 | - | - | 1.25 | 1.33 | - | - |
| monks-1 | 3.70 | - | 3.18 | 2.95 | - | - | 2.41 | 2.58 | - | - | 2.05 | 2.19 | - | - |
| monks-2 | 6.00 | - | 6.00 | 6.00 | - | - | 6.00 | 6.00 | - | - | 6.00 | 6.00 | - | - |
| audiology.stand | 13.40 | 8.85 | 9.53 | 9.73 | 8.98 | 9.38 | 4.87 | 5.02 | 9.59 | 10.28 | 4.70 | 5.12 | 7.43 | 7.79 |
| soybean-large | 5.23 | 15.50 | 5.15 | 4.73 | 6.95 | 7.03 | 3.20 | 3.30 | 7.01 | 7.02 | 2.63 | 2.65 | 5.44 | 5.72 |
| house-votes-84 | 3.47 | - | 3.38 | 3.27 | - | - | 2.35 | 2.15 | - | - | 1.64 | 1.43 | - | - |
| breast-cancer-w | 2.05 | - | 2.14 | 2.12 | - | - | 1.33 | 1.29 | - | - | 0.74 | 0.72 | - | - |
| semeion | 9.51 | 16.12 | 9.10 | 9.18 | 13.61 | 13.57 | 7.40 | 7.39 | 10.52 | 10.48 | 6.12 | 6.07 | 8.49 | 8.49 |
| dna (splice) | 5.95 | 6.75 | 5.43 | 5.35 | 5.65 | 5.60 | 4.25 | 4.22 | 4.21 | 4.18 | 3.68 | 3.54 | 3.71 | 3.55 |
| kr-vs-kp (chess) | 14.64 | - | 11.05 | 10.42 | - | - | 8.17 | 7.34 | - | - | 5.41 | 5.48 | - | - |
| optdigit-recogn | 3.26 | 4.85 | 3.13 | 3.15 | 4.23 | 4.20 | 2.55 | 2.46 | 3.29 | 3.21 | 2.08 | 2.10 | 2.71 | 2.65 |
| pendigit-recogn | 2.00 | 3.03 | 2.00 | 2.00 | 2.53 | 2.60 | 1.61 | 1.57 | 2.08 | 2.11 | 1.38 | 1.35 | 1.86 | 1.74 |
| agaricus-lepiot | 1.29 | - | 1.88 | 2.09 | - | - | 1.11 | 1.11 | - | - | 0.60 | 0.60 | - | - |
| nursery | 2.81 | 5.03 | 2.80 | 2.64 | 4.57 | 4.47 | 2.14 | 2.17 | 3.48 | 3.41 | 1.85 | 1.90 | 2.83 | 2.92 |
| letter-recogn | 3.98 | 6.83 | 3.85 | 3.83 | 6.08 | 5.98 | 3.69 | 3.53 | 4.89 | 5.01 | 2.83 | 2.85 | 4.33 | 4.32 |
| # iterations | 1 | | 20 | | | | 50 | | | | 100 | | | |
| soybean-small | 2.03 | 2.82 | 0.72 | 0.72 | 1.04 | 1.07 | 0.64 | 0.64 | 0.93 | 1.04 | 0.59 | 0.60 | 0.90 | 0.91 |
| heart-spect | 6.90 | - | 1.21 | 1.20 | - | - | 0.50 | 0.46 | - | - | 0.27 | 0.20 | - | - |
| zoo | 3.10 | 5.22 | 1.44 | 1.51 | 1.93 | 1.99 | 1.32 | 1.38 | 1.68 | 1.78 | 1.27 | 1.33 | 1.63 | 1.71 |
| promoters | 3.75 | - | 0.35 | 0.38 | - | - | 0.13 | 0.15 | - | - | 0.07 | 0.06 | - | - |
| monks-3 | 2.98 | - | 0.95 | 0.73 | - | - | 0.40 | 0.43 | - | - | 0.17 | 0.21 | - | - |
| monks-1 | 3.70 | - | 1.79 | 1.96 | - | - | 2.41 | 1.30 | - | - | 1.72 | 1.68 | - | - |
| monks-2 | 6.00 | - | 6.00 | 6.00 | - | - | 6.00 | 6.00 | - | - | 6.00 | 6.00 | - | - |
| audiology.stand | 13.40 | 8.85 | 3.91 | 4.23 | 6.85 | 7.57 | 3.04 | 3.45 | 5.85 | 6.58 | 2.61 | 3.15 | 4.75 | 5.81 |
| soybean-large | 5.23 | 15.50 | 2.29 | 2.32 | 4.96 | 5.09 | 1.93 | 1.98 | 4.18 | 4.35 | 1.82 | 1.86 | 3.54 | 3.73 |
| house-votes-84 | 3.47 | - | 0.88 | 0.89 | - | - | 0.38 | 0.36 | - | - | 0.18 | 0.19 | - | - |
| breast-cancer-w | 2.05 | - | 0.41 | 0.34 | - | - | 0.15 | 0.14 | - | - | 0.08 | 0.07 | - | - |
| semeion | 9.51 | 16.12 | 5.15 | 5.12 | 7.20 | 7.17 | 4.52 | 4.50 | 6.60 | 6.61 | 4.33 | 4.31 | 6.41 | 6.44 |
| dna (splice) | 5.95 | 6.75 | 3.16 | 3.06 | 3.28 | 3.25 | 2.81 | 2.75 | 2.96 | 2.90 | 2.66 | 2.60 | 2.76 | 2.72 |
| kr-vs-kp (chess) | 14.64 | - | 4.35 | 4.04 | - | - | 2.00 | 1.79 | - | - | 1.00 | 1.01 | - | - |
| optdigit-recogn | 3.26 | 4.85 | 1.88 | 1.92 | 2.48 | 2.46 | 1.70 | 1.76 | 2.20 | 2.29 | 1.62 | 1.68 | 2.18 | 2.28 |
| pendigit-recogn | 2.00 | 3.03 | 1.23 | 1.22 | 1.56 | 1.53 | 1.14 | 1.13 | 1.40 | 1.40 | 1.09 | 1.10 | 1.35 | 1.35 |
| agaricus-lepiot | 1.29 | - | 0.30 | 0.27 | - | - | 0.13 | 0.12 | - | - | 0.06 | 0.06 | - | - |
| nursery | 2.81 | 5.03 | 1.79 | 1.90 | 2.64 | 2.77 | 1.72 | 1.83 | 2.50 | 2.58 | 1.72 | 1.78 | 2.42 | 2.48 |
| letter-recogn | 3.98 | 6.83 | 2.64 | 2.63 | 3.89 | 3.79 | 2.47 | 2.47 | 3.43 | 3.45 | 2.42 | 2.44 | 3.28 | 3.31 |

sification models. On one hand we can traditionally consider both techniques as classification methods. On the other hand rough sets and approximate reducts provide methodology to analyze dependencies in the data, reduce its dimensionality and construct models that are easy understandable and follow well known principles e.g. Occam's razor [5]. Approximate reducts ensembles provide additional layer in analyzing these dependencies, yet keep the model simple. We showed how approximate reducts can be easily integrated into boosting algorithm. It is very interesting to analyze what rules apply during reduct calculation when object weights are changed in iterative manner as in AdaBoost algorithm. We tested our methods on different benchmark datasets but we did not focus much on

parameter setup. We will focus in future on finding methods to tune these parameters, but as more important task, we also find to search for methods to automatically adjust them. For example what is the best level of approximation to start with, or should it be adjusted during training procedure. We are also interested in minimizing number of iterations necessary to build good quality classification models.

References

1. Jan G. Bazan and Marcin S. Szczuka. The Rough Set Exploration System. In James F. Peters and Andrzej Skowron, editors, *Transactions on Rough Sets III*, volume 3400 of *Lecture Notes in Computer Science*, pages 37–56. Springer, 2005.
2. Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.
3. Thomas G Dietterich. Ensemble methods in machine learning. In *Multiple classifier systems*, pages 1–15. Springer, 2000.
4. Thomas G. Dietterich and Ghulum Bakiri. Solving multiclass learning problems via error-correcting output codes. *arXiv preprint cs/9501101*, 1995.
5. Pedro Domingos. The role of occam’s razor in knowledge discovery. *Data mining and knowledge discovery*, 3(4):409–425, 1999.
6. Yoav Freund. Boosting a weak learning algorithm by majority. *Information and computation*, 121(2):256–285, 1995.
7. Yoav Freund, Robert Schapire, and N Abe. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612, 1999.
8. Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. In Lorenza Saitta, editor, *Machine Learning, Proceedings of the Thirteenth International Conference (ICML ’96), Bari, Italy, July 3-6, 1996*, pages 148–156. Morgan Kaufmann, 1996.
9. Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.
10. Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.
11. Peter Sollich Anders Krogh. Learning with ensembles: How over-fitting can be useful. In *Proceedings of the 1995 Conference*, volume 8, page 190, 1996.
12. Ludmila I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley, 2014.
13. Ludmila I. Kuncheva, Juan José Rodríguez Díez, Catrin O. Plumptre, David E. J. Linden, and Stephen J. Johnston. Random Subspace Ensembles for fMRI Classification. *IEEE Transactions on Medical Imaging*, 29(2):531–542, 2010.
14. Ludmila I. Kuncheva and Christopher J. Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, 51(2):181–207, 2003.
15. M. Lichman. UCI machine learning repository, 2013.
16. Mikhail Ju Moshkov, Marcin Piliszczyk, and Beata Zielosko. *Partial covers, reducts and decision rules in rough sets: theory and applications*, volume 145. Springer, 2009.
17. Sinh Hoa Nguyen, Jan Bazan, Andrzej Skowron, and Hung Son Nguyen. Layered learning for concept synthesis. In *Transactions on Rough Sets I*, pages 187–208. Springer, 2004.
18. Zdzislaw Pawlak and Andrzej Skowron. Rudiments of Rough Sets. *Information Sciences*, 177(1):3–27, 2007.
19. Robi Polikar. Ensemble based systems in decision making. *Circuits and Systems Magazine, IEEE*, 6(3):21–45, 2006.
20. Robert E Schapire and Yoram Singer. Improved boosting algorithms using confidence-rated predictions. *Machine learning*, 37(3):297–336, 1999.

21. Dominik Slezak. Normalized Decision Functions and Measures for Inconsistent Decision Tables Analysis. *Fundamenta Informaticae*, 44(3):291–319, 2000.
22. Dominik Slezak. Rough Sets and Functional Dependencies in Data: Foundations of Association Reducts. In Marina L. Gavrilova, Chih Jeng Kenneth Tan, Yingxu Wang, and Keith C. C. Chan, editors, *Transactions on Computational Science V*, volume 5540 of *Lecture Notes in Computer Science*, pages 182–205. Springer, 2009.
23. Dominik Slezak and Sebastian Widz. Is it important which rough-set-based classifier extraction and voting criteria are applied together? In *Rough Sets and Current Trends in Computing*, pages 187–196. Springer, 2010.
24. Dominik Slezak and Sebastian Widz. Rough-set-inspired feature subset selection, classifier construction, and rule aggregation. In *Rough sets and knowledge technology*, pages 81–88. Springer, 2011.
25. Sebastian Stawicki and Sebastian Widz. Decision bireducts and approximate decision reducts: Comparison of two approaches to attribute subset ensemble construction. In *Computer Science and Information Systems (FedCSIS), 2012 Federated Conference on*, pages 331–338. IEEE, 2012.
26. Roman W. Swiniarski and Andrzej Skowron. Rough set methods in feature selection and recognition. *Pattern Recognition Letters*, 24(6):833–849, 2003.
27. Sebastian Widz and Dominik Slezak. Rough set based decision support - models easy to interpret. In *Rough Sets: Selected Methods and Applications in Management and Engineering*, pages 95–112. Springer, 2012.
28. Sebastian Widz and Dominik Slezak. Attribute subset quality functions over a universe of weighted objects. In *Rough Sets and Intelligent Systems Paradigms*, pages 99–110. Springer, 2014.
29. Jakub Wroblewski. Adaptive aspects of combining approximation spaces. In *Rough-Neural Computing*, pages 139–156. Springer, 2004.
30. Ji Zhu, Hui Zou, Saharon Rosset, and Trevor Hastie. Multi-class adaboost. *Statistics and its Interface*, 2(3):349–360, 2009.

WZMACNIANIE KLASYFIKATORÓW OPARTYCH NA PRZYBLIŻONYCH REDUKTACH

Streszczenie. W niniejszej pracy rozważono konstrukcję grup klasyfikatorów konstruowanych w oparciu o przybliżone redukt decyzyjne i teorię zbiorów przybliżonych. Szczególną uwagę poświęcono reduktom wyliczanym na zbiorach danych, w których każdemu obiektowi przyporządkowana jest pewna waga (wartość rzeczywista). Pokazano, w jaki sposób takie redukty mogą zostać użyte do konstrukcji grup klasyfikatorów w oparciu o algorytmy wzmocniania tzw. słabych klasyfikatorów (ang. boosting). Przeanalizowano dwa podejścia do wyszukiwania reduktów. Nasze metody przetestowano na danych benchmarkowych.

Słowa kluczowe: wzmocnianie klasyfikatorów, zbiory klasyfikatorów, zbiory przybliżone, aproksymacyjne nieredukowalne zbiory atrybutów, wybór podzbiorów cech

IBS PAN *serie*

47 789

Bibl. podręczna

ISBN 83-894-7558-8