Feature Set-based Consistency Sampling in Bagging Ensembles

Jerzy Błaszczyński[1], Roman Słowiński[1,2], Jerzy Stefanowski[1]

Institute of Computing Science, Poznań University of Technology[1],
60-965 Poznań, Poland
Systems Research Institute, Polish Academy of Sciences[2],
01-447 Warsaw, Poland
{jurek.blaszczynski, roman.slowinski, jerzy.stefanowski}@cs.put.poznan.pl

Abstract. In this paper, we study the influence of changing the technique of bootstrap sampling on the classification performance of bagging ensembles. In standard bagging each training object has the same probability of being selected to bootstrap sample. We propose a feature set-based consistency sampling, where the local information about class distribution in the objects' neighborhood is used to produce bootstrap samples more focused on consistent objects. We use random feature set selection to determine the neighbourhood. This sampling technique may lead to ensembles learned on slightly more diversified bootstrap samples with more accurate component classifiers. The experiments show that proposed technique leads to improvement in the global performance of tree and rule bagging ensembles.

1 Introduction

Bagging is one of the most popular approaches for constructing ensembles of classifiers. Its idea is quite simple and effective. The ensemble is a set of so called component or base classifiers induced by the same learning algorithm on a number of bootstrap samples drawn from the training set. The outputs of component classifiers are combined in equal weight voting [3]. Classification performance of bagging results mainly from perturbation in random subsamples of the training data and combining of component classifiers in ensemble [17].

Bagging procedure [3] has been extended in a number of ways in attempt to improve the classification accuracy of the ensemble. These extensions focused mainly on increasing diversity of component classifiers. *Random forest* is a well known example of such extension. It uses feature subset randomized decision tree component classifiers [6]. Other extensions of bagging profit from random selection of features. In some cases, several random subspaces of features are selected along with the idea presented as the *random subspaces* method [15]. In other cases, the random selection of features is combined with standard bootstrap sampling. Examples of such ensembles of classifiers were considered by different researchers [18, 19, 24].

In this paper, we consider a new extension of bagging that is touching another aspect of creating bootstrap samples. Let us remind that in the standard bootstrap sampling the equal probability of drawing is assigned to each object from the training set. We postulate a modification of distribution of the samples that results from the different procedure of drawing objects. This procedure focuses on analyzing local properties in the training data. The motivation is that not all objects in the training data are equally important for induction of accurate classifiers. Our research hypothesis is that one may use information about the objects' neighborhood in sampling. More precisely, we consider class distribution of objects' neighbors. This allows us to recognize some as *consistent* objects (i.e., objects safer for predicting class label because their neighbors have the same or similar descriptions by attributes and the same class label). Objects are distinguished as *inconsistent* when their neighbors that have similar descriptions are labeled by different classes. The inconsistent objects are usually located in boundaries between classes or in noisy sub-regions of the problem space. These objects, if not treated appropriately by the learning method, may lead to overfitting and decrease classification performance of the standard, single classifiers. Moreover, depending on the number of neighbors with the same class label and the number of neighbors with different class label one can consider degrees of consistency.

We can pose a research question whether it is possible to slightly change bootstrap sampling in such a way that component classifiers are less influenced by inconsistent objects. At the same time, we want the ensemble of classifiers to benefit from the distribution of consistent objects. Following from this motivation, we postulate a higher diversity in bootstrap samples with respect to inconsistent objects (i.e., more diversified bootstrap samples in this respect). We measure the consistency of each object in its neighborhood and use this value to estimate the probability of drawing the object into sample. In this way, we modify the standard bootstrap sampling with uniform probability distribution into more focused distribution where consistent objects are more likely to be selected than inconsistent ones. The goal is to learn component classifiers on more perturbed distributions characterised by higher rates of consistent objects.

This modification is definitely different to stronger changes in input data made by some pruning techniques typical for single classifiers which may omit difficult, noisy objects. Here, we do not remove such objects from the samples. They can be still selected to bootstrap samples but with the lower probability. Moreover, the probability of selection is controlled by choice of consistency measure. In our view, the decrease of the probability of selecting inconsistent or noisy objects in the bagging scheme may lead to creating more accurate and still sufficiently diversified component classifiers.

Moreover, we would like to stress that this modification is also different to the idea of boosting where an iterative identification of incorrectly classified objects is made. Boosting consists in subsequent extending of an ensemble of classifiers by adding component classifiers focused on objects incorrectly classified so far. In our approach, we evaluate consistency of objects and we change their sampling probability in a *pre-processing phase* before learning of component classifiers. This is different than evaluating objects' classification in boosting. Additionally, in the way typical for bagging, consistency of objects is calculated independently for each of bootstrap samples.

Our first contribution is to introduce consistency sampling technique in bagging scheme. Our previous experiments with consistency sampling on full sets of features [2] indicate that it allows to increase classification ability of classifiers on highly inconsistent data. Here presented technique can be viewed as a kind of two level bagging. On the first level, we use a preprocessing method that assigns a weight to each of training objects. The weight reflects the value of consistency measure calculated for this object with regard to its neighborhood and its class label. In general, the neighborhood of the object can be modelled in different ways. However, in the current paper, we use discretization of continuous features and identify neighbors belonging to the same classes/blocks of discretization subintervals. This method seems natural since some kind of handling of continuous attributes similar to internal discretization is applied anyways in the learning phase of tree and rule base classifiers that we use in experiments. Moreover, the neighborhood of the object is selected on the basis of a random subset of features of a given size. The use of feature subsets may increase the number of similar neighbors. It also follows from intuitive premise saying that if an object keeps sufficient consistency even for smaller sets of features, it could be a good candidate for a seed to generalize its description for well supported decision rule or branch in decision tree. More consistent objects are assigned higher weights. On the second level of our approach the weights are used in the bagging scheme. The feature sets are used only for calculation of the weights. The sampling of objects takes into account their complete description.

The consistency of objects is measured on the basis of class labels distribution in its neighborhood by means of either *rough membership function* [26] or *monotonic measure of consistency* [1]. These measures are quite simple and fast to calculate and reflect natural properties of objects' distribution in the training data. They have already been used for learning rules [1].

The other aim of this paper is to carry out a comprehensive experimental study on several benchmark data sets, where we evaluate the usefulness of the new proposal of bagging constructed with either decision tree or rule based component classifiers. We compare them against the standard version of bagging to study the influence of the modified sampling on the classification accuracy.

In the next section, we remind bagging scheme and we show some other related works. In section 3, we define consistency measures and give their interpretation. In section 4, we describe feature set-based consistency sampling. Section 5 brings presentation of learning algorithms used to create base classifiers and experimental setup. In the following section 6, results of experiments are presented and discussed. We conclude by giving remarks and recommendations for applications of presented techniques.

2 Related Works

As the main aim of our research is to extend bagging we present an overview of its standard version and we briefly discuss the related extensions which use feature selection or specific sampling of training data.

2.1 Bagging

The *Bagging* approach introduced by Breiman [3] is based on the concepts of manipulating input data by bootstrap sampling and then combining predictions of classifiers. The bootstrap sample is obtained by uniformly sampling with replacement objects from the training set. Given the parameter k, which is the number of repetitions or component classifiers, k different bootstrap samples $S_1, S_2, \ldots, S_k$ are generated. From each new training sample S_i a classifier C_i is induced by the same learning algorithm and the final classifier C^* is formed by aggregating k classifiers. A final classification of object x is built by a uniform voting scheme on $C_1, C_2, \ldots, C_k$, i.e. is assigned to the class predicted most often by these component classifiers, with ties broken arbitrarily. For more details see, e.g., [3, 17].

Many experimental results show a significant improvement of the classification accuracy, in particular, using decision tree classifiers. An improvement is also observed when using rule classifiers [23]. However, the choice of a base classifier is not indifferent. According to Breiman [3], what makes a base classifier suitable is its *unstability.* A base classifier is unstable, when small changes in the learning set cause major changes in the classifier. For instance, the decision tree and rule classifiers are unstable, while K-Nearest Neighbor classifiers are not. For more theoretical discussion on the justification of the problem why bagging works the reader is referred to [3, 17].

Let us come back the key concept of using several perturbed training sets. Each *bootstrap sample* is obtained by uniformly sampling with replacement objects from the original learning set. So, some objects do not appear in it, while others may appear more than once. Let the original training set consist of m objects. Each sample contains $n \leq m$ objects (usually it has the same size as the original set). The same probability $1/m$ of being sampled is assigned to each object. The probability of an object 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 objects from the original learning set [3].

Thus, on average, approximately $36,8\%$ of objects from the original training set are not present in a given bootstrap sample. Following some motivations from the first section we could suspect that some bootstrap samples may contain less misleading training objects than the complete original training set. Consequently more accurate classifiers could be generated and aggregating them may improve classification performance.

Let us also remark that the bagging is a kind of parallel algorithm in training and classification phases, i.e., there is no transfer of additional information between components unlike in the boosting which iteratively builds a new classifier using information about performance of the previously generated base classifiers.

2.2 Adaptive Resampling and Combining

To extend the last remark from the previous section we would like to notice that Breiman refers to methods that can improve accuracy of unstable classifiers by perturbing and combining. The key concept of the P & C method (perturb and combine) is to generate multiple versions of the classifier by perturbing the training set and then to combine these multiple versions into a single classifier. Breiman proposed some P & C methods for bagged classifiers. Among them are *Arcing Classifiers* [4], *Pasting Small Votes* [5] and *Random Forests* [6]. In particular these methods perturb the training data by sequentially sampling with replacement objects, where at each step probability of selecting a given object is modified by its importance. This importance is estimated at each step by the accuracy of a new base classifier.

The reader familiar with ensembles classifiers can notice other solutions to taking into account misclassification of training objects. In boosting [21] more focus is given on objects difficult to classify.

2.3 Feature Subsets in Bagging

The most powerful extension of bagging with decision trees is *Random forest* introduced by Breiman [6]. It is a kind of generalization of bagging where each tree classifier is additionally randomized. So, besides using bootstrap sampling of objects from the training set, for each node of the tree a subset of p features from the original set of r features is randomly selected. The tree induction algorithm selects the best split on p as the new node of the tree. In general, such a randomization should increase diversity of component classifiers. Breiman suggested to grow in this way unpruned CART tree and showed in experiments that it significantly improved the classification performance comparing to other ensembles, see details in [6, 17]. He also recommended to select $(log_2(r) + 1)$ of features.

Another approach to increase diversity of component classifier with selecting feature subsets comes from inspirations of the *Random Subspace Method* [15]. In this method each component classifier in the ensemble is constructed using a different randomly chosen subset of features (so training sets are diversified by using different feature subsets). Ho [15] suggested that good results were obtained for selecting $r/2$ features, where r is the number of all features in the original training set.

Latine at al. [18] proposed combining bootstrap bagging with a random feature selection. In this approach k bootstrap samples of objects are generated as in the standard bagging. Then, for each sample, r subset of features (of size p) are randomly selected. Thus, one gets $r \times k$ new training sets and apply the learning algorithm to construct classifiers in the ensemble. Authors [18] showed that such a combination of decision trees performed better than Random Subspace Method and standard bagging. Quite similar approach was also considered in [19] - where more extensive experimental evaluation was carried out. Using other less random features selection in this combination was also studied in [24]. To sum up, we can repeat after [17] that combination of bagging and feature selection aims at making the ensemble more diverse than using each of these methods alone.

3 Measures of Consistency

Consistency measures operate on sets of similar objects that are called neighbours of the considered object $\mathbf{x}$. In this paper we restrict interest on qualitative and discrete descriptions – so sets of neighbours are a kind of *elementary classes of relation (elementary sets)* constructed with respect to available information about values of features. Let us define these sets more formally. Given a set P of measured values of features / attributes characterizing properties of an object $\mathbf{x} = (x_1, x_2, \dots, x_n)$, an equivalence relation, or *indiscernibility* between objects is

$$\mathbf{x} =_P \mathbf{x}' \Leftrightarrow x_i = x'_i \qquad \forall i \in P.$$

We can define neighbourhood (or set) of indiscernible objects as such that $\forall \mathbf{x}, \mathbf{x}'$:

$$N_P^{=}(\mathbf{x}) = \{\mathbf{x}, \mathbf{x}' | \mathbf{x} =_P \mathbf{x}'\}. \tag{1}$$

If object $\mathbf{x}$ belongs to neighbourhood $N_P^{=}$, in which all objects are assigned the same output value / decision class X_i, $i = 1, \dots, t$ then $\mathbf{x}$ is consistent.

Indiscernibility relation is a natural choice for discrete data with limited attributes' domains. For continuous domains, discretization of objects' description [8] is usually needed to effectively use this relation.

Having different numbers of objects with the same value of a decision class in the neighbourhood $N_P^{=}$, object x is consistent to different degree which can be measures by appropriate consistency measures. Below two chosen measures are presented.

Rough membership, called also μ consistency measure, of object $\mathbf{x}$ with respect to a decision class X_i is defined [26] as

$$\mu_{X_i}^{P}(\mathbf{x}) = \frac{|N_P^{=}(\mathbf{x}) \cap X_i|}{|N_P^{=}(\mathbf{x})|}, \tag{2}$$

where $|\cdot|$ denotes cardinality. Rough membership captures a ratio of number of objects in neighbourhood $N_P^{=}(\mathbf{x})$ and in considered set X_i, to number of all objects present in the neighbourhood $N_P^{=}(\mathbf{x})$. This measure is an estimate of conditional probability $Pr(\mathbf{x} \in X_i | \mathbf{x} \in N_P^{=}(\mathbf{x}))$. It can be shown that estimation of this probability by frequencies, as it is done in (2), is equivalent to the maximum likelihood estimation under the assumption of common class probability distribution of objects within each neighbourhood $N_P^{=}(\mathbf{x})$.

The next, ϵ consistency measure, $\epsilon_{X_i}^{P}(\mathbf{x})$ is defined [1] as

$$\epsilon_{X_i}^{P}(\mathbf{x}) = \frac{|N_{\overline{P}}^{=}(\mathbf{x}) \cap \neg X_i|}{|\neg X_i|}. \tag{3}$$

In the numerator of (3), is the number of objects in the training data that belong to neighborhood $N_{\overline{P}}^{=}(\mathbf{x})$ and do not belong to set X_i. The ratio $\epsilon_{X_i}^{P}(\mathbf{x})$ is an estimate of conditional probability $Pr(\mathbf{x} \in N_{\overline{P}}^{=}(\mathbf{x}) | \mathbf{x} \in \neg X_i)$, called also a catch-all likelihood [9]. Probability $Pr(\mathbf{x} \in N_{\overline{P}}^{=}(\mathbf{x}) | \mathbf{x} \in \neg X_i)$ can be rewritten as $\frac{Pr(\mathbf{x} \in N_{\overline{P}}^{=}(\mathbf{x}) \wedge \mathbf{x} \in \neg X_i)}{Pr(\mathbf{x} \in \neg X_i)}$. Logically, implication $\mathbf{x} \in N_{\overline{P}}^{=}(\mathbf{x}) \rightarrow \mathbf{x} \in X_i$ can be rewritten as $\neg(\mathbf{x} \in N_{\overline{P}}^{=}(\mathbf{x}) \wedge \mathbf{x} \in \neg X_i)$. Thus, the intuition of calculating measure $\epsilon_{X_i}^{P}(\mathbf{x})$ is that we can see how far the implication, i.e., rule, stating that $\mathbf{x}$ belongs to X_i is not supported by objects in neighbourhood of $\mathbf{x}$.

To use measures $\mu_{X_i}^{P}(\mathbf{x})$ and $\epsilon_{X_i}^{P}(\mathbf{x})$ in consistency sampling we need to transform them to measure $c^P(\mathbf{x})$ defined for a given object $\mathbf{x}$ as

$$c^P(\mathbf{x}) = \mu_{X_i}^{P}(\mathbf{x}) \quad \text{or} \quad c^P(\mathbf{x}) = 1 - \epsilon_{X_i}^{P}(\mathbf{x}), \tag{4}$$

where X_i is the class label of object x.

4 Feature Set-based Consistency Sampling

The goal of feature set-based consistency sampling is to increase the global predictive accuracy of bagged classifiers by using additional local information that reflects consistency of objects with regard to subsets of their features. The resulting bagged classifiers are trained on bootstrap samples slightly shifted towards more consistent objects.

The learning algorithm presented as Algorithm 1 is almost the same as the standard bagging scheme. The difference lies in feature set-based consistency sampling, the procedure of bootstrap sampling that is used in line 3. The rest of the bagging scheme remains unchanged. In feature set-based consistency sampling, when sampling with replacement from the training set is performed, a measure of consistency $c^P(\mathbf{x})$ is calculated for each object $\mathbf{x}$ from the training set. A consistent object $\mathbf{x}$ has $c^P(\mathbf{x}) = 1$, inconsistent objects have $0 \leq c^P(\mathbf{x}) < 1$. The consistency measure is used to tune the probability of object x being sampled to a bootstrap sample, e.g. by calculating a product of $c^P(\mathbf{x})$ and $1/m$; where m is the number of objects in training data. Thus, objects that are inconsistent have decreased probability of being sampled. Objects that are more consistent (i.e., have higher value of a consistency measure) are more likely to appear in the bootstrap sample. Different measures of consistency may result in different probability of inconsistent object $\mathbf{x}$ being sampled. The value of $c^P(\mathbf{x})$ involving ϵ measure is usually higher than $c^P(\mathbf{x})$ involving μ measure. As it comes from formula (3), ϵ measure relates the number of inconsistent objects to the whole number of objects in the data set that may cause inconsistencies. So, for large data sets the value of ϵ measure may be relatively high for all objects. On the other hand, from (2), μ measures inconsistency more locally. It relates the number of consistent objects in the neighborhood to the number of objects in the neighborhood.

It is worth noting that, feature sets are used only to calculate consistency of objects. Objects with complete description are drawn into bootstrap samples and then used during learning of component classifiers.

Algorithm 1: Feature set-based consistency sampling in bagging scheme

```
Input  : $LS$ learning set;
         $TS$ testing set;
         $LA$ learning algorithm;
         $c$ consistency measure;
         $p$ number of features used in consistency sampling;
         $k$ number of bootstrap samples;
Output: $C^*$ final classifier

Learning phase;
for $i := 1$ to $k$ do
    $S_i$ := bootstrap sample of objects having complete description; objects are
    drawn by consistency sampling from $LS$ with measure $c$ calculated on
    randomly selected $p$ features {sample with replacement according to
    measure $c$ } ;
    $C_i := LA(S_i)$ {generate a base classifier} ;
end
Classification phase;
foreach $\mathbf{x}$ in $TS$ do
    $C^*(\mathbf{x}) = \arg\max_X \sum_{i=1}^{T} (C_i(\mathbf{x}) = X)$ {the class with maximum number of
    votes is chosen as a final label for $\mathbf{x}$} ;
end
```

We should stress the difference of sampling in boosting (and other adaptive resampling and combining methods) to the consistency-based sampling presented in this work. In general, boosting develops the ensemble of classifiers by subsequent addition of a new component classifier to the ensemble. This component classifier is trained on a sample which is drawn from the original data set according to the performance of the ensemble of classifiers. The objects on which the ensemble of classifiers performed poorly are more likely to be drawn into the sample on which the new component classifier is learned. The presented here bagging with consistency sampling is not an incremental, stepwise approach and does not change sampling towards objects that pose difficulty for the ensemble. Our extension of sampling is definitely different to boosting idea.

5 Experimental Setup

The main aim of experiments presented in this paper is to evaluate the usefulness of our modification of sampling objects into bootstraps in improving the classification accuracy of bagging. We consider two variants: the first, where sampling is modified by evaluating μ consistency measure, and the second, where it is modified by ϵ consistency measure. We compare these variants of bagging against the standard bagging using the same number of component classifiers to study how much these modifications may improve the classification performance of bagging. The magnitude of improvement may depend on used consistency measure. To reduce the effect of too high values of ϵ consistency measure discussed in section 4, we transformed the value of ϵ measure by the exponential function of high order (to be more specific, we used values of $(\epsilon^P(\mathbf{x}))^{128}$ in experiments; we do not expect significant difference in results unless the order of the exponential function is not significantly lower). Predictions of classifiers inside the bagging were always aggregated into the final classification decision by equal weight voting.

In all compared versions of bagging the component classifiers were generated by the same learning algorithm. We decided to compare decision trees or rules ensembles, because they are accurate but unstable which is good bagging. Such kind of classifiers may be particularly influenced by the possible inconsistency of data [2]. Moreover, rule induction algorithms following sequential covering principle correspond to discovery of local patterns in data.

Therefore, we selected the well known Quinlan C4.5 algorithm [20] and run it with standard parameters except generating unprunned decision trees. This follows Breiman's recommendations for using tree classifiers inside bagging scheme. As the rule induction algorithms we chose two options. Namely, PART algorithm [10] and MODLEM algorithm . The last choice resulted from three motivations: it has been already successfully applied inside few multiple classifiers [23]; the classification ability of the classifier built on its sets of rules is usually comparable to C4.5 rules or tree classifier; moreover it has been previously shown that it could be used with pre-processing of inconsistent objects [22, 25].

As it is not so well known as other rule induction algorithms, we briefly remark that the MODLEM algorithm is also based on the idea of a *sequential covering* and it generates an *unordered minimal set* of decision rules for every decision concept. The main procedure for rule induction scheme starts from creating a first rule by choosing sequentially the best elementary conditions according to chosen criteria (in our experiments we used an entropy based one). When the rule is stored, all learning positive objects that match this rule are removed from consideration. The process is repeated while some significant positive examples of the decision concept remain still uncovered. For inconsistent data a kind of pruning strategy can be used - although in our experiments we used unpruned version to be consistent with the other algorithms. More detailed description of this algorithm can be found in [14, 25]. We use classification strategy for solving ambiguous, multiple or partial matches proposed in [11] to classify objects with rules induced by MODLEM. This strategy takes into account coverage of all rules completely matched and also allows partial matching if no rule fits the description of the new object.

The number of component classifiers in each bagging ensemble was set to 20. This number led to comparable performance of bagging ensembles when compared to other perturb and combine methods in our previous experiments.

Additionally, we study consistency of objects in bootstrap samples and similarity between bootstrap samples. We check average values for 1000 bootstrap samples created by standard bagging, and bagging extended by feature set-based consistency sampling with μ measure and ϵ measure.

We evaluated performance for 14 data sets listed in Table 1. They come mainly from the UCI repository[1]. We chose them because they were often used by other researchers working with rule ensembles. Several of these data sets included numerical attributes so we used discretization method to effectively use indiscernibility relation to select neighbors (see section 3). We used a well known supervised method based on minimizing class entropy [8], in the version which automatically determine the necessary number of cut points for each of the attributes.

Table 1. Characteristics of data sets

Data set	Objects	Attributes	Classes
breast-w	699	9	2
bupa	345	6	2
credit-german	1000	20	2
crx	690	15	2
diabetes	768	8	2
ecoli	336	7	8
glass	214	9	7
heart-cleveland	303	13	5
hepatits	155	19	2
ionosphere	351	34	2
pima	768	8	2
sonar	208	60	2
vehicle	846	18	4
vowel	990	13	11

6 Results of Experiments and Discussion

The classification accuracy was estimated by the stratified 10-fold cross-validation, which was repeated several times. Tables with results always contain the average classification accuracy and the standard deviation of classification

[1] see http://www.ics.uci.edu/˜mlearn/MLRepository.html

accuracy. Moreover, we include the rank of the average classification accuracy calculated for all variants of classifiers and the given data set. The rank is presented in brackets (the smaller rank, the better).

We compare standard bagging to bagging extended by feature set-based consistency sampling with μ and ϵ consistency measures. We checked two sizes of the feature sets used to measure the consistency of objects. First, we chose 50% of original feature set size following recommendation given in [15]. Then, we also checked ln of original feature set size following [6]. Because there were no huge differences in results, we present results for 50% feature set size (which were better) and summarize results for ln feature set size. The rank given in a tables 2, 3 and 4 reflects position of the average classification accuracy of a 50% random feature ensemble when compared to all other variants in 50% random feature ensembles. We show these ranks because they are used in further described statistical test. Last row of each table shows the average rank scored by a given type of 50% random feature ensemble.

Table 2. Classification accuracy in repeated 10-fold cross validation of an ensemble of 20 C4.5 classifiers standard bagging and feature set-based consistency bagging on 50% of features. Rank of the results presented in Tables 2, 3 and 4 is given in brackets.

	C4.5		
data set	std. bagging	μ bagging	ϵ bagging
breast-w	95.61±0.07 (9)	95.76±0.5 (8)	96.04±0.3 (6.5)
bupa	54.4±0.5 (8)	56.33±0.1 (1)	53.53±0.8 (9)
credit-g	71.33±0.9 (9)	72.4±0.08 (7)	71.9±0.3 (8)
crx	83.72±0.9 (7)	85.27±0.5 (3)	85.6±0.07 (1)
diabetes	77.52±0.2 (7.5)	78.04±0.5 (4)	77.95±0.2 (5.5)
ecoli	84.03±0.1 (6)	84.42±0.4 (3)	83.53±0.7 (7)
glass	76.48±0.4 (4)	76.79±0.8 (3)	70.25±1 (8)
heart-c	78.1±2 (9)	81.41±2 (7)	82.73±0.4 (5)
hepatitis	80.65±2 (8)	79.57±1 (9)	81.08±1 (7)
ionosphere	91.45±0.4 (9)	91.83±0.5 (6.5)	91.83±0.4 (6.5)
pima	77.52±0.2 (7.5)	78.04±0.5 (4)	77.95±0.2 (5.5)
sonar	75±0.4 (8)	75±0.4 (8)	75±0.4 (8)
vehicle	71.16±0.3 (9)	71.4±0.8 (8)	71.51±0.6 (7)
vowel	85.05±0.2 (7)	84.68±0.2 (9)	84.88±0.2 (8)
average rank	7.71	5.75	6.57

We use a statistical approach to compare difference in performance between classifiers in variants which we mentioned above. First, we apply Friedman test to globally compare performance of nine different classifiers (i.e., C4.5, MODLEM and PART in standard bagging, μ bagging and ϵ bagging) on multiple data sets [7, 16]. The null-hypothesis in this test is that all compared classifiers perform equally well. It uses ranks of each of classifiers on each of the data sets. The

Table 3. Classification accuracy in repeated 10-fold cross validation of an ensemble of 20 MODLEM classifiers standard bagging and feature set-based consistency bagging on 50% of features. Rank of the results presented in Tables 2, 3 and 4 is given in brackets.

	MODLEM		
data set	std. bagging	μ bagging	ϵ bagging
breast-w	96.23±0.1 (3.5)	96.23±0.3 (3.5)	96.09±0.2 (5)
bupa	55.27±1 (4)	55.17±0.5 (5)	55.85±1 (2)
credit-g	74.5±1 (4)	74.6±0.6 (3)	74.87±0.3 (1)
crx	83.33±0.2 (8.5)	84.5±0.4 (6)	85.36±0.2 (2)
diabetes	78.3±0.4 (1)	78.12±0.3 (3)	78.21±0.5 (2)
ecoli	80.26±0.6 (9)	82.94±0.6 (8)	84.13±0.4 (5)
glass	66.67±1 (9)	73.36±0.4 (5.5)	72.74±2 (7)
heart-c	81.52±1 (6)	84.05±1 (1)	83.72±1 (2)
hepatitis	81.72±2 (6)	83.01±2 (5)	83.23±1 (4)
ionosphere	92.21±0.4 (5)	92.3±0.6 (4)	91.64±0.1 (8)
pima	78.3±0.4 (1)	78.12±0.3 (3)	78.21±0.5 (2)
sonar	80.77±1 (2)	80.77±1 (2)	80.77±1 (2)
vehicle	73.84±0.7 (1)	73.09±0.5 (3)	73.4±0.4 (2)
vowel	89.09±0.3 (3)	89.6±0.2 (2)	89.87±0.3 (1)
average rank	4.5	3.86	3.21

Table 4. Classification accuracy in repeated 10-fold cross validation of an ensemble of 20 PART classifiers standard bagging and feature set-based consistency bagging on 50% of features. Rank of the results presented in Tables 2, 3 and 4 is given in brackets.

	PART		
data set	std. bagging	μ bagging	ϵ bagging
breast-w	96.04±0.2 (6.5)	96.66±0.2 (1)	96.52±0.4 (2)
bupa	54.49±0.2 (7)	55.65±1 (3)	54.88±1 (6)
credit-g	72.97±0.6 (6)	73.83±0.6 (5)	74.73±0.6 (2)
crx	83.33±0.8 (8.5)	84.88±0.5 (5)	84.98±0.07 (4)
diabetes	77.52±0.3 (7.5)	77.95±0.6 (5.5)	77.39±0.2 (9)
ecoli	84.52±0.5 (2)	85.32±0.6 (1)	84.23±0.2 (4)
glass	77.41±0.9 (2)	77.88±1 (1)	73.36±1 (5.5)
heart-c	80.2±0.7 (8)	83.17±0.3 (4)	83.28±1 (3)
hepatitis	85.38±0.8 (1)	83.66±2 (3)	84.73±2 (2)
ionosphere	92.4±0.4 (3)	92.6±0.7 (2)	93.16±1 (1)
pima	77.52±0.3 (7.5)	77.95±0.6 (5.5)	77.39±0.2 (9)
sonar	79.33±0.7 (5)	79.33±0.7 (5)	79.33±0.7 (5)
vehicle	72.58±0.3 (5)	72.42±0.5 (6)	72.62±1 (4)
vowel	88.42±0.3 (6)	88.72±0.2 (5)	88.82±0.4 (4)
average rank	5.36	3.71	4.32

lower rank, the better classifier. Friedman statistics for these results gives 5.3

which exceeds the critical value 2.03 (for confidence level 0.05). We have not presented complete post-hoc analysis [7] of differences between classifiers. However, we show the average ranks of each of classifiers in tables. The results of Friedman test and observed differences in average ranks between classifiers allow us to state that there is a significant difference between them.

We continue our comparison with examination of importance of difference in classification performance between each pair of classifiers. However, we are more focused on differences between the same classifier used in different variants of bagging. We apply Wilcoxon test [16] with null-hypothesis that the medians of results on all data sets of the two compared classifiers are equal. Let us remark, that in the paired tests ranks are assigned to the value of difference in the average classification accuracy between compared pair of classifiers. When we apply this test to results of C4.5 classifiers, it detects statistically important difference in pairs between standard bagging and μ bagging (p-value around 0.03). In case of MODLEM, p-value in Wilcoxon test comparing difference between standard bagging and ϵ bagging is equal 0.0503. On the other hand, p-value in test comparing MODLEM in standard bagging with MODLEM in μ-bagging is much higher, around 0.1. Let us notice that MODLEM gives the best results in standard bagging. Thus, relatively, it is the hardest case for improvement. In case of examining PART results, statistically important difference is found between standard bagging and μ bagging (p-value around 0.02).

The results of feature set-based consistency bagging with ln feature set size also indicate a statistically important difference in Friedman test (p-value in this test is around 0.008). However, the difference of ranks between classifiers in this test are smaller. Moreover, Wilcoxon test applied for pairs of classifiers does not allow us to distinguish statistically important differences for most of the pairs as it gives higher p-values.

We study consistency of objects in bootstrap samples and similarity between samples as a continuation of the presented comparison of the accuracy. The purpose of this study is to show differences between sampling used in standard bagging and 50% feature set-based consistency sampling with μ measure and ϵ measure. The results are presented in Table 5. In the study, any object $\mathbf{x}$, for which $c^P(\mathbf{x}) < 1$ is considered as inconsistent. First, we check the average percentage of inconsistent objects in bootstrap samples. This shows the fraction of inconsistent objects in bootstrap samples created by compared versions of bagging. Then we check the average consistency of an object drawn into sample. This allows us to compare the average probability of object being drawn into the samples. Finally, we compare in pairs all bootstrap samples created by each of versions of bagging. We check the similarity of all objects in samples and similarity of inconsistent objects in samples. We define similarity for a pair of bootstrap samples as the value of the ratio of the sum of the same objects drawn into the samples to the number of objects in the samples. Then we calculate similarity for a given version of bagging as the average value of similarity of all pairs of samples created by the version of bagging. This allows to compare how diversified are bootstrap samples created by each version of bagging. Moreover, we can check the diversity of the samples with respect to all objects and only with respect to the inconsistent objects.

When we compare the average percentage of inconsistent objects from Table 5, we can see that samples drawn by feature set-based consistency sampling are more consistent than those drawn by standard bagging. Moreover, we can see that ϵ sampling leads to lower average percentage of inconsistent objects in samples (with one exception of vowel data set). Less numerous data set described by fewer attributes have higher percentages of inconsistent objects (see results for bupa, diabetes, ecoli, glass and pima). The average consistency of sample (i.e., the average probability of object being drawn in the sample), is lower for these data sets than for others. One data set, sonar, is not affected at all by feature set-based consistency sampling. This is the data set described by the highest number of attributes. Thus, we can see that feature set-based consistency sampling is sensitive to the number of features used to calculate consistency of objects.

Similarity of bootstrap samples created by standard bagging is always close to 0.75 regardless of whether it is calculated for all objects or for inconsistent ones. We treat this result as a base line for our comparison. We can see that similarity measured for all objects drawn in bootstrap samples created by feature set-based consistency sampling is higher then for standard bagging. On the other hand, in most of the cases, the similarity measured with respect to inconsistent objects is lower than in standard bagging. The exceptions to this rule are previously mentioned small data sets. For these data sets, similarity between samples increase when we calculate it for all objects and inconsistent objects. We relate these observations to the classification accuracy results presented in tables 2, 3 and 4. It becomes quite apparent that in most of the cases, feature set-based consistency sampling increases the accuracy when similarity of samples with respect to inconsistent objects decreases. Moreover, the accuracy tends to be higher when similarity of samples calculated for all object does not increase greatly.

7 Conclusions

In this paper we considered an extension of bagging where consistency of objects on a subset of features is taken into account while drawing objects into bootstrap samples. Two measures of objects' consistency were compared: μ rough membership and ϵ monotonic measure. We calculate these measures on the basis of distribution of class labels in the given objects' neighborhood which was identified with using discretization in random feature subspaces of a given size.

Results of experiments showed that the new sampling improved the classification accuracy of bagging. However, the range of this improvement depends on a few aspects. The highest increase when comparing standard bagging and μ bagging was observed for C4.5 trees. On the other hand, one should notice that the accuracy of the standard version of bagging with trees was lower than versions with rule sets. The statistical comparison of results showed that the best of our extended variants of bagging may be different depending on the used

Table 5. Consistency and similarity of bootstrap samples created by standard bagging and 50% feature set-based consistency sampling with measures μ and ϵ

data set	type of sampling	% inconsistent objects	avg. consistency	similarity all	similarity inconsistent
breast-w	standard	18.38	-	0.75	0.755
	μ	16.31	0.98	0.755	0.71
	ϵ	13.38	0.95	0.760	0.57
bupa	standard	88.82	-	0.751	0.75
	μ	85.08	0.62	0.751	0.725
	ϵ	49.2	0.64	0.892	0.814
credit-german	standard	41.38	-	0.75	0.749
	μ	33.09	0.89	0.768	0.63
	ϵ	27.07	0.89	0.802	0.602
crx	standard	40.66	-	0.751	0.75
	μ	33.16	0.91	0.768	0.648
	ϵ	26.2	0.89	0.809	0.582
diabetes	standard	96.63	-	0.751	0.751
	μ	95.59	0.72	0.771	0.768
	ϵ	86.52	0.34	0.856	0.847
ecoli	standard	93.44	-	0.75	0.75
	μ	91.49	0.8	0.788	0.789
	ϵ	80.81	0.54	0.901	0.882
glass	standard	85.58	-	0.751	0.749
	μ	80.04	0.72	0.788	0.742
	ϵ	55.94	0.62	0.894	0.807
heart-c	standard	62.45	-	0.751	0.753
	μ	56.36	0.86	0.768	0.697
	ϵ	35.87	0.77	0.857	0.528
hepatits	standard	32.84	-	0.752	0.75
	μ	27.27	0.94	0.762	0.661
	ϵ	4.12	0.97	0.883	0.01
ionosphere	standard	1.41	-	0.75	0.74
	μ	1.37	0.99	0.751	0.699
	ϵ	0.55	0.99	0.753	0.01
pima	standard	96.63	-	0.751	0.751
	μ	95.59	0.72	0.771	0.768
	ϵ	86.52	0.34	0.856	0.847
sonar	standard	0	-	0.752	0
	μ	0	1.0	0.751	0
	ϵ	0	1.0	0.751	0
vehicle	standard	51.96	-	0.75	0.751
	μ	41.73	0.85	0.778	0.641
	ϵ	37.99	0.86	0.815	0.668
vowel	standard	46.94	-	0.75	0.75
	μ	32.74	0.85	0.79	0.575
	ϵ	33.25	0.87	0.801	0.639

component classifier. Feature set-based consistency bagging with μ measure is the best choice for C4.5 and PART algorithms. The variant using ϵ measure is a better choice in case of MODLEM as a rule component classifier. These observations are strongly supported by the analysis of average ranks and results of Wilcoxon test. We may attribute this difference in performance of classifiers in μ consistency bagging and in ϵ consistency bagging to different strategy of constructing component classifiers. Since rules generated in PART results from induction of a tree, C4.5 and PART are tree based algorithms. MODLEM, on the other hand, is a pure rule induction algorithm based on sequential covering technique. The results of experiments indicate that μ consistency sampling works for tree based learning methods while ϵ consistency sampling works for rule learning methods. This hypothesis should be, however, further investigated.

We compared classification accuracy results of presented here modifications to the results of Random Forest with the same number of component decision tree classifiers. Random Forest gives comparable results to bagging extended by feature set-based consistency sampling on the data sets that we used in experiments. We plan a more comprehensive comparison as the future work.

We observed that the number of randomly selected features cannot be too small. The differences between feature set-based consistency bagging and standard bagging were more visible for random sub-samples of 50% size of the original feature sets than for choosing ln of this size. The influence of the size of feature set used for consistency bagging should be more deeply examined in the future.

Comparison of similarity between samples drawn in feature set-based consistency sampling and standard uniform bootstrap sampling shown that the feature set-based consistency sampling enables to construct more diversified samples with respect to inconsistent objects.

Finally, as a future research we should point to studying other methods of selecting objects' neighborhood. In particular, we should examine appropriate measures of distance between objects.

References

1. Błaszczyński, J., Greco, S., Słowiński, R., Szeląg, M.: Monotonic Variable Consistency Rough Set Approaches. International Journal of Approximate Reasoning (2009) doi: 10.1016/j.ijar.2009.02.011.
2. Błaszczyński, J., Słowiński, R., Stefanowski, J.: Variable consistency bagging ensembles. Transactions on Rough Sets, (accepted for publication 2009; to appear).
3. Breiman, L.: Bagging predictors. Machine Learning, 24 (2) (1996) 123–140.
4. Breiman, L.: Arcing Classifiers. The Annals of Statistics, 26 (3) (1998) 801–824.
5. Breiman, L.: Pasting small votes for classification in large databases and on-line. Machine Learning, 36 (1999) 85–103.
6. Breiman, L.: Random Forests. Machine Learning, 45 (1) (2001) 5–32.
7. Demsar, J.: Statistical Comparisons of Classifiers over Multiple Data Sets. Journal of Machine Learning Research 7 (2006) 1–30.
8. Fayyad U. M., Irani K. B.: Multi–interval discretization of continuous-valued attributes for classification learning, Proceedings of 13th Int. Conf. on Machine Learning, Morgan Kaufmann, 1993, 1022–1027.

9. Fitelson, B.: Likelihoodism, Bayesianism, and relational confirmation. Synthese, vol. 156, Springer Netherlands, 2007, pp. 473–489.
10. Frank E., Witten I. Generating Accurate Rule Sets Without Global Optimization. Proceedings of the Fifteenth International Conference on Machine Learning (1998), 144 - 151.
11. Grzymala-Busse, J. W.: Managing uncertainty in machine learning from examples. Proc. 3rd Int. Symp. in Intelligent Systems (1994) 70–84.
12. Grzymala-Busse, J.W.: A new version of the rule induction system LERS. *Fundamenta Informaticae* 31 (1997) 27–39.
13. Grzymala-Busse J.W., Zou X.: Classification strategies using certain and possible rules, In Proceedings of the RSCTC'98 Conference, Lecture Notes in Artificial Intelligence 1424, Springer Verlag (1998) 37–44.
14. Grzymala-Busse, J. W., Stefanowski, J.: Three approaches to numerical attribute discretization for rule induction. International Journal of Intelligent Systems, 16 (1) (2001) 29–38.
15. Ho, T.K.: The random subspace method for constructing decision forests. IEEE Transactions on pattern analysis and machine intelligence, 24 (2) (1998) 832–844.
16. Kononenko, I., Kukar, M.: Machine Learning and Data Mining. Horwood Pub. 2007
17. Kuncheva, L.: Combining Pattern Classifiers. Methods and Algorithms. Wiley (2004).
18. Latinne, P., Debeir, O., Decaestecker, Ch.: Different Ways of Weakening Decision Trees and Their Impact on Classification Accuracy of Decision Tree Combination. In: Proc. of the 1st Int. Workshop of Multiple Classifier Systems, Springer Verlag LNCS 1857, (2000).
19. Panov, P., Dzeroski S.: Combining bagging and random subspaces to create better ensembles. Proceedings IDA 2007 Conference, Springer LNCS 4723 (2007) 118–129.
20. Quinlan J.R., C4.5: Programs for Machine Learning, Morgan Kaufmann, 1992.
21. Schapire, R. E., Singer, Y.: Improved boosting algorithms using confidence-rated predictions, Machine Learning 37 (1999), no. 3, 297–336.
22. Stefanowski J.: The rough set based rule induction technique for classification problems. Proc. of 6th European Conference on Intelligent Techniques and Soft Computing EUFIT 98, (1998) 109–113.
23. Stefanowski J.: The bagging and n2-classifiers based on rules induced by MODLEM. Proceedings of RSCTC'04 Conference, Springer LNAI 3066 (2004) 488–497.
24. Stefanowski J.,Kaczmarek M.: Integrating attribute selection to improve accuracy of bagging classifiers. Proc. of the AI-METH 2004 Conference - Recent Developments in Artificial Intelligence Methods, Gliwice, 2004, 263-268.
25. Stefanowski, J.: On combined classifiers, rule induction and rough sets. In: Peters J. et al. (eds): Transactions on Rough Sets VI, LNCS, vol. 4374, Springer (2007) 329–350.
26. Ziarko W.: Variable precision rough sets model. Journal of Computer and Systems Sciences 46 (1) (1993) 39–59.