An Experimental Study of Methods Combining Multiple Classifiers - Diversified both by Feature Selection and Bootstrap Sampling

Jerzy Stefanowski

Institute of Computing Science, Poznań University of Technology,
ul. Piotrowo 3A, 60–965 Poznań, Poland,
Jerzy.Stefanowski@cs.put.poznan.pl

Abstract. Ensemble approaches are learning algorithms that construct a set of classifiers and then classify new instances by combining their predictions. These approaches can outperform single classifiers on wide range of classification problems. In this paper we proposed an extension of the bagging classifier integrating it with feature subset selection. Moreover, we examined the usage of other methods for integrating answers of these sub-classifiers, in particular a dynamic voting instead of simple voting combination rule. The extended bagging classifier (with induced decision trees as base sub-classifiers) was evaluated in an experimental comparative study with standard approaches.

Keywords: machine learning, supervised classification, bagging, feature selection, dynamic integration of classifiers.

1 Introduction

Machine learning is a domain intensively developed in last decades, see e.g. [19, 20]. One of its main sub-domain is *supervised learning*, where given a set of *learning examples* of an *unknown target function*, described by a set of *features*, the learning algorithm has to discover this function. If values of the target function are drawn from a discrete set of classes, it is a case of *classification* and the learning algorithm outputs a *classifier*. It can be further applied to predict classes of new objects. It should noticed that the similar problem is also considered in statistical learning or data mining [6], although machine learning generally aims on developing algorithms which automatically improve their performance with analysing experience and offers rather more symbolic knowledge representation [19].

Nowadays the most active research in supervised learning includes an *integration* of several base classifiers into the combined classification system [4, 13, 32]. Such systems are known under the names *multiple classifiers, ensembles methods, committees* or *classifier fusion*. This topic attracts an interest of machine learning or statistics researchers as multiple classifiers are often much more accurate than the component classifiers that make them up.

Many approaches for constructing multiple classifiers have been developed - for good reviews the reader can look, e.g., [4, 9, 13, 26, 32]. The most successful approaches include manipulating the learning set (as it is done in boosting and bagging), manipulating the input features, using different learning algorithms to the same data, manipulating the output targets. The component classifiers are typically combined by voting. The *diversification* of these classifiers is a necessary condition for their efficient combination [4, 32]. In this paper we will consider only these diversification methods that manipulate input data: either by sampling of learning examples or by feature selection. In this way several different learning sets are obtained from the input data set and then the same learning algorithm is run over them. The popular method for sampling examples is *bagging* [2], which uses bootstrap sampling, while selected feature subsets for ensembles could be obtained in many ways (although random search is the most popular - see the review in section 3). There are many experimental or theoretical papers demonstrating that these methods lead to achieving a higher classification accuracy than single classifiers.

However, the literature study shows that these two diversification dimensions are considered independently. The open research question is - whether integrating both techniques of manipulating input learning data, i.e. bootstrap sampling and selection of multiple feature subsets, will also allow us to achieve better classification accuracy than the sole solution. Thus, the main aim of this paper is an experimental study of the effectiveness of the enhanced approach. According to our best knowledge the most related work to this task is the study by Lattine *et al.* [16], where they introduced such a mixed approach called *BagFs*, although the features were just randomly selected several times. Their experimental results showed that *BagFs* never performed worse or even performed better on some data sets than the standard bagging. In our paper, we would like to consider enhancing this proposal by integrating with more advanced methods of feature selection than plain random drawing of features only. We will take into account different methods evaluating the relationship between each feature, or feature subsets, and the target class.

Moreover, we want to put the other question - whether the simple equal weight voting is a sufficient combination rule for our enhanced bagging. As noticed by some researchers, see e.g. [29, 30], it is important to have a good integration method that utilize the diversity of component sub-classifiers. If some sub-classifiers are more accurate in some sub-spaces of the input domain but may be inaccurate on the rest of it, it could be beneficial to promote their decisions for these objects which they are better specialized for. In particular, previous research with feature selection only showed the usefulness of some strategies, which dynamically change votes, while aggregating predictions of base classifiers depending on the description of the classified object, or select the most accurate classifiers [29]. Thus, the other aim of this paper is to experimentally verify the usefulness of different methods for integrating the answers of sub-classifiers in the proposed enhancement of the bagging. In this sense the current paper extends our previous paper on the similar topic, which was focused more on the feature selection [28].

The evaluation is based on many comparative experiments performed on a diverse collection of machine learning benchmark data sets [1]. In all experiments the sub-classifiers are *decision trees* induced by the Ross J. Quinlan $C4.8$ algorithm. For its and other methods implementations we used libraries available inside *Weka* software [33] extended by programming our own necessary libraries.

2 Combining Classifiers

According to Dietterich's presentation [4], in the standard supervised learning problem, a learning algorithm is given a set of learning examples of the form $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$ for some unknown function $y = f(x)$. The $\mathbf{x}_i$ values are feature being vectors of the form $< x_{i,1}, \ldots, x_{i,n} >$, where m is the size of the training set and n is the number of features. Given a set of training examples $\mathbf{T}$, a learning algorithm outputs a classifier. Then, for a new object $\mathbf{x}$, it predicts the unknown value y.

In general, an ensemble consists of s component, base classifiers, denoted as $C_1, \ldots, C_s$. During a construction phase, each base classifier is trained using learning examples of the sets $\mathbf{T}_1, \ldots, \mathbf{T}_s$. For each example or a new object, the predicted outputs of each of these base classifiers are combined in some way $C^* = F(C_1, \ldots, C_s)$ to produce the final classification decision of the ensemble.

Previous theoretical research (see, e.g., their summary in [4, 9, 32]) indicated that combining several classifiers is effective only if there is a substantial level disagreement among them, i.e. they make error independently with respect to one another, and the error rate of each base classifier should not exceed certain limit. Combining identical classifiers (i.e. making identical / correlated errors) is useless.

As a result, methods for creating ensembles focus on producing diversified base classifiers. Numerous techniques try to manipulate the training set. In this paper we have chosen *Bagging*, which was introduced by Breiman [2]. It is the most straightforward way of manipulating the learning set and is based on bootstraping sampling with replacement. Each sample has the same size as the original set, however, some examples do not appear in it, while others may appear more than once. For a training set with m examples, the probability of an example being selected at least once is $1-(1-1/m)^m$. For a large m, this is about 1 - $1/e$. According to Breiman [2] each bootstrap sample contains 63.2% unique examples from the training set.

A family of bootstrap samples $(\mathbf{T}_1, \ldots, \mathbf{T}_s)$ from the original learning set $\mathbf{T}$ is obtained. From each sample T_i a classifier C_i is induced by the same learning algorithm and the final classifier C^* is formed by aggregating T classifiers. A final classification of object x is built by an *equal voting scheme* on $C_1, C_2, \ldots, C_T$, i.e. the object is assigned to the class predicted most often by these sub-classifiers. The bagging is shortly presented in the below procedure. For more details see [2].

Experimental results presented in [2, 4, 16, 18, 24, 26, 27] show a significant improvement of the classification accuracy. However, the choice of the learning method is not indifferent. This method works especially well for *unstable* learning algorithms - i.e. algorithms whose output classifier undergoes major changes in response to small changes in the learning data. For instance, the decision tree, artificial neural networks and rule learning algorithms are unstable, while K-Nearest Neighbor classifiers or linear threshold algorithms are not as they are very stable. For more theoretical discussion on the bagging the reader is referred to [2, 6].

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

3 Ensemble Feature Selection - Related Works

The feature subset selection is an important problem in machine learning, knowledge discovery or statistical literature [3, 11, 17, 19]. Typically, this problem is referred to the single learning algorithm and the aim is to find the subset of features leading to at least similar classification accuracy than the set of all features. It is usually a difficult problem as it involves *searching* a potentially huge *space.* The selection of the attribute subset is often based on a given *evaluation measure.* Such measures usually evaluate a degree of relationship between values of a single feature and a decision class. The typical search strategy evaluates each feature on its own and then selects a subset with the highest ranked features. Other measures are appropriate for evaluating subsets of features. Here, the search strategy is often stepwise, where in each iteration it is tried to add (in so called forward search) the most promising feature or (in backward search) to remove the less important one, if such operation results in a receiving a better subset. For reviewing different methods of feature selection, see e.g. [17]. In general could be useful for:

1. creating better classifiers by removing redundant and irrelevant features;
2. handling high dimensionality in some data mining problems;
3. knowledge discovery - i.e. determine what features are and are not influential in weak theory domains.

However, it should be stressed that within the context of ensembles the motivation for feature subset selection is different. Feature subset selection is used as a mechanism for introducing the *diversity of base classifiers.* According to it, the learning sets for creating the ensemble, i.e. $(\mathbf{T}_1, \ldots, \mathbf{T}_s)$, are obtained by using different subsets of feature for each of them. Finding a set of feature subsets for constructing an ensemble havaing accurate and diverse base classifiers is also known under the name *ensemble feature selection* [21]. One can have a look to [32, 13] for a review of these approaches.

Ho [7] has shown in his *Random Subspace Method* (RS) that random selection of feature subsets may be an effective technique because the lack of accuracy is the ensemble members is compensated by their diversity. In RS, one randomly selects a given proportion k of features from the original set of features. The aggregation is usually performed using weighted voting on the basis of the base classifier accuracy. There are also other variants of RS, where different numbers of features are randomly selected instead of the fixed proportion.

In other approaches, the correlation between each feature and the output of the class is computed and the base classifier is trained only on the most correlated feature subsets. Yet other methods partition the set of features in such a way that each subset is used by one classifier - this occurs for some applications, especially in text or speech recognition. Recently some researches used genetic algorithm to get feature subsets optimizing both accuracy and diversity of base classifiers [31, 25]. The discussion and experimental study of partitioning the feature space using different combination schemes presented in [14] led to conclusions that there is no one best feature combination for all situations.

4 Methods for Integrating Answers of Classifiers

Another important issue in creating an ensemble is the choice of the function for combining the predictions of the base classifiers. As it is discussed in literature, if the integration method does not properly utilize the ensemble diversity, then no benefit arises from integrating multiple models [13, 30].

In general, there are two kinds of methods: *group combination* or *specialized selection* [9, 13, 30]. In the first method all base classifiers are consulted to classify a new object while the other method chooses only these classifiers whose are "expertised" for this object.

Voting is the most common method used to combine predictions of single classifiers. In its simplest version, called a *equal weighted* voting, the classification prediction of each base classifier is considered as an equally weighted vote for the particular class. The class that receives the highest number of votes is selected as the final classification. Often, the vote of each classifier may be *weighted*, e.g., by the estimating its accuracy of the corresponding classifier. Other group approaches use a *Bayesian decision rule*, where assuming mutual independence between classifiers one selects the class with the highest posterior probability. In situations where the classifiers outputs as fuzzy supports for the class, fuzzy aggregation methods are also applied starting from using the simpler aggregation operators as *Minimum, Maximum, Product* and *Ordered Weight Averaging* [12] up to possibilistic *Dempster-Schafer* combination rule.

Another idea consists in *explicitly training a combination rule* - usually *a second level meta-learning algorithm* is put on the outputs of base

classifiers and has to learn a correct final answer of the system from their predictions. The meta-combiner is usually based on the concepts of meta-classifiers or stacked generalization - for more details see [9, 29].
A number of *selection* methods have also been proposed, for review see e.g. [29, 31]. In a case of bagging or feature ensembles the *dynamic integration* methods are often used. In [30] three techniques called *Dynamic Selection, Dynamic Voting* and *Dynamic Voting with Selection* were considered. All these are based on local accuracy estimates. During the learning phase, the learning set is partitioned by the cross validation and accuracy of each sub-classifier is estimated for each learning example. When a new example is provided for classification, first its nearest neighbours (examples) are found in the learning set using a distance metric based on its feature values. Then, the classification accuracies of all the sub-classifiers on the neighbours set are calculated. In *Dynamic Voting* all of the sub-classifiers are used in a weighted voting, each with a weight proportional to its accuracy. *Dynamic Selection* chooses the subset of classifiers with the highest classification accuracy to produce the final decision. According to [29] the above methods led to a slightly better accuracy than the simple Equal Weight Voting for both bagging and boosting classifiers. In other experiments with ensemble feature selection both dynamic voting and selection work significantly better than weighted voting or simple aggregation rules [30, 31].

5 Integrating Feature Selection, Bagging and Dynamic Selection of Classifiers

The previously discussed approaches attempt to obtain uncorrelated subclassifiers constructing them either by example sampling or by feature subset selection. However, both these diversification techniques are considered independently. The research question pointed in section 1 has concerned the advantage of taking into account both techniques together. The simplest integration schema includes the typical bagging with Random Subspace Method. First s bootstrap samples T_i of the learning set are generated from the original set T (with the same sampling as in [2]). Then, in each set one additionally independently samples R subsets of f' features - they are selected from the f initial ones without replacement ($k = f'/f$ is the same proportion for all subsets using a uniform probability distribution). Such an approach has already been considered in [15]; more details on a heuristic procedure for determining k are also given there. Finally, one obtains $S \cdot R$ new learning sets to which the learning algorithm is applied. Latine at al considered such an approach, called *BagFs*, for Quinlan decision-tree induction and the final prediction of this system followed the original bagging - computing the majority class from all $S \cdot R$ base predictions. They performed an experimental evaluation of their approach comparing it against the standard bagging and the multiple feature selection only [16] and the results of these experiments showed that this approach never performed worse or even performed better on some data sets than the other models combining the same number of base classifiers. The statistical analysis also showed that if the proportion of selected features is not too large, and it was able to exhibit the high level if diversity between its components and offered the highest degree of accuracy.

In our enhanced approach we would like to select features according to more complex methods than plain random choice only. We want to include other methods evaluating in the different way the relationship between each feature or feature subsets and the decision class. We propose to replace in *BagFs* the R random feature selection iterations by R feature selection iterations, each conducted according to another evaluation measure. Thus, the base sub-classifiers could be trained on the more classification relevant subsets of features. However, by choosing different methods we also want to have diversified, multiple subsets. In [28] we have already studied the problem of choosing such methods. We performed an experiment on data sets, where each selection method was applied to bootstrap samples obtained by the standard bagging. Every constructed bagging classifier consisted of 10 bootstrap. Due to the size of the paper we skip the detailed results and summarize that finally the we chosen the search method using the following evaluation measures:

– *Contextual-merit measure*: Proposed in [8] evaluates single features not their subset. It assigns the highest merit to features, where examples from different values classes have different values. Its definition is also presented in [22].
– *Info-Gain* : The known measure based on the information entropy often used in symbolic induction [20].
– *Chi-Squared statistic*: It is based on widely used statistics to evaluate pairs of features. Any numeric feature have to be discretized [33].
– *Correlation-based measure*: The idea behind it is that a good subset should contain features highly correlated with the class but uncorrelated with each other, see [5].

As the last method we considered the *wrapper approach* [11], where the search algorithm conducts a forward stepwise search for a subset of features using the classifier itself as the evaluation function (by calculating a classification accuracy obtained by this classifier).

The first three methods evaluate the single featuresand the choice of features is done according to their ranking - which requires the parameter k for the best features (i.e. the percentage number of features to retain). To establish this value (only one value for each data set) we used an approach from [16], where k was experimentally tuned by performing a nested cross validation evaluation for the bagging with only random feature selection. The following thresholds were used for the data sets (see section 6): glass 40%, bupa 70%, vote 60%, breast 20%, election 30%, wine 30%, ecoli 60% and german 80%. The same values were used for all studied selection methods. The last two methods evaluate the feature subsets.

Additionally, we want to use a different integration methods to aggregate the answers of sub-classifiers in the proposed enhancement of the bagging. As we want to compare the usefulness of different methods, the following one will be experimentally verified:

– Simple Equal Weight Voting,

- Stacked Meta-Combiner - which was implemented as a decision tree induced by C4.5 algorithm
- Dynamic Voting (see the description in section 4)

In dynamic voting we compute nearest learning examples of the classified object with an Euclidean distance measure for numeric features and simplified Value Difference Metric for symbolic ones.

6 Experiments

The aim is to experimentally verify the usefulness of the proposed enhanced approach integrating different feature subset selection methods with the bagging classifier and to evaluate the impact of applying the different methods of integrating answers of sub-classifiers.

In all experiments the base sub-classifiers are decision trees induced by the C4.8 algorithm available in WEKA [33]. We used standard options of this algorithm. Unless otherwise stated the decision tress are built as prunned. The classification accuracy was estimated by the 10-fold stratified cross validation technique. All the results in tables are presented as an average classification accuracy with a standard deviation. When performance of two classifiers on the same data will be compared we will use a paired t-Student test with the significant level equal 0.05.

We used 8 following data sets: *glass, bupa, vote, breast cancer Wisconsin, bush-election, wine, ecoli, german.* The data sets were chosen in such a way, that they have different number of features of particular types, different number of examples and there are some data sets with two-class distribution and some with more than two classes. All the data except *bush-election* are coming from UCI repository [1]. *Bush-election* comes from Hadjmichael and Wasilewska study. *Bupa* (also known as *liver-disorder*) and *vote* used in this thesis are shortened versions of the original data sets.

First, we had to decide about the configuration, i.e. the number of bootstrap samples. In [15] *Bagging* and *BagFs* was constructed with different numbers of bootstraps - either 49 or 343. The results for a higher number of samples were not significantly better. As learning time tended to be long for too many samples and from previous studies with the simple bagging we knew that the smaller number of bootstraps is often sufficient (see [24, 27]), we checked different configurations with a smaller number of components. To be more precise we took into consideration few variants: (1) the standard bagging (with 49 bootstraps, called Bag_{49} from now) and 7-bootstrap-7-random-chosen-feature-subsets $BagFs$ (called Bag_7Fs_7); (2) the standard bagging with 25 bootstraps Bag_{25} and Bag_5Fs_5. The results for both variants were similar with small advantages for the variant (1) - for 2 data sets *bupa* and *ecoli* the differences were statistically significant (details are given in [10]. The number of base classifiers (49) was also chosen in order to make results of our experiment easily comparable with results of [16].

Moreover, we wanted to check also different number of feature selection iterations - to see what could be more important: increasing the number of bootstrap samples or increasing the number of feature selection iterations. In [10] we tested different combinations of samples and feature subsets iterations, e.g. Bag_5Fs_5, Bag_7Fs_7, Bag_5Fs_{10}, Bag_4Fs_{12}, $Bag_{10}Fs_5$, $Bag_{12}Fs_4$, $Bag_{16}Fs_3$. Some of these results are summarized in table 1. The results were compared in pairs by T - student paired test with a general conclusion that increasing the number of bootstraps slightly improves classification accuracy while increasing the number of feature iterations with maintaining the number of bootstraps seems to be meaningless.

Table 1. Different configurations of Bagging with feature selection – a comparison of classification accuracies (an average value with a standard deviation represented in %).

Data	C4.8	Bag_{49}	Bag_5Fs_{10}	Bag_7Fs_7	$Bag_{10}Fs_5$	$Bag_{12}Fs_4$
glass	67.76±1.44	74.77±1.62	76.45±1.78	77.01±1.80	76.03±1.62	77.29±2.2
bupa	65.42±1.21	73.62±0.85	70.26±1.3	71.91±1.81	72.96±1.59	72.46±1.9
vote	94.23±0.65	94.80±0.28	94.77±0.4	94.83±0.39	94.97±0.4	94.97±0.4
breast	94.48±0.62	96.25±0.39	94.36±0.5	94.56±0.75	94.55±0.76	95.01±0.82
election	90.56±0.66	91.22±0.76	91.8±0.5	92.23±0.66	92.45±0.63	92.73±0.48
wine	93.82±1.18	96.07±0.88	94.72±1.4	95.00±1.44	94.83±1.32	95.72±1.22
ecoli	83.10±1.04	84.38±0.80	83.81±1.4	84.61±0.74	84.55±0.9	84.64±1.2
german	69.22±1.30	74.14±0.88	72.82±1.5	73.65±1.12	74.13±0.9	73.62±0.98

Further on, we compared the performance of the proposed feature selection approaches. In all cases we used the simplest equal weight voting as an integration method. As we wanted to construct a multiple classifier as similar in a structure as Bag_7Fs_7 and Bag with 49 repetitions, we always used around 50 components. We started with 10 bootstrap samples and, then, for each of these samples 5 iterations of chosen feature subsets selection methods were applied - this version is denoted as $Bag_{10}DFS_5$. We verified other versions of this approach using only 4 feature selection iterations for each bootstrap (in this case the classifier used 12 bootstraps - more details in [28]). They were obtained by removing one feature selection method, e.g., $Bag_{12}DFS_4 - Corr$ – was a classifier consisting of 12 bootstraps and 4 feature selection iterations for each bootstraps, each of the 4 iterations used a different measure: Contextual Merit, Info-Gain, Chi-Squared statistic and Plain Random drawing. In table 2 we present a comparison of classification accuracies for some variants of these $BagDFS$ classifiers. The observations from table 2 were following: Bag_{49} and Bag_7Fs_7 (except *breast-cancer*) were significantly better than the single $C4.8$ classifier. The differences between Bag_{49} and Bag_7Fs_7 depend on the data. Comparing $Bag_{10}DFS_5$ against $BagFS$ we observed generally a similar accuracy. The difference depends on the data set. The most accurate variant is $BagDFS$ variant without considering Info-Gain and Chi-Squared statistics. We also verified whether this configuration performs better, when the base classifier decision trees are unprunned – the classification results are generally comparable with slightly superiority in favour of unprunned version, in particular for *election* data the difference is significant.

Table 2. $BagDFS$: several variants comparison.

Data	Bag_{10}	$Bag_{12}DFS_4$		$Bag_{16}DFS_3$
set	DFS_5	$-Corr$	$-IG$	$-IG-Chi$
glass	76.87 ±2.22	77.48 ±1.46	77.06 ±1.52	76.54 ±1.9
bupa	70.32 ±1.64	70.99 ±1.28	71.13 ±2.11	71.39 ±1.13
vote	94.97 ±0.11	94.80 ±0.28	95.00 ±0.05	95.00 ±0.1
breast	95.99 ±0.38	96.09 ±0.39	96.11 ±0.25	96.07 ±0.36
election	90.95 ±0.85	91.35 ±0.85	91.42 ±0.80	91.98 ±0.75
wine	96.69 ±1.04	95.96 ±1.42	97.02 ±1.06	97.08 ±0.96
ecoli	83.99 ±1.44	83.81 ±0.94	83.78 ±0.90	83.80 ±0.89
german	74.25 ±1.03	73.95 ±0.96	74.43 ±0.47	74.58 ±0.59

Table 3. Equal weight voting, stacked combination vs. dynamic voting comparison.

Data	$Bag_{16}DFS_3$	$Bag_{16}DFS_3$	$Bag_{16}DFS_3$	$BagFS$
set	$+EV$	$+DV$	$+SC$	$+DV$
glass	76.54±1.9	76.87±1.87	68.71±2.33	76.26±1.18
bupa	71.39±1.13	71.16±1.0	66.81±1.41	71.74±2.04
vote	95.0±0.1	95.0±0.1	94.40±0.16	94.77±0.64
breast	96.07±0.36	96.18±0.22	95.26±0.44	96.44±0.34
election	91.98±0.75	92.50±0.53	90.95±0.7	92.39±0.52
wine	97.08±0.96	97.08±1.02	93.31±1.28	96.74±0.37
ecoli	83.80±0.89	83.86±0.91	80.77±1.46	83.51±0.43
german	74.58±0.59	74.79±0.61	71.97±1.2	73.29±1.08

In the last experiments we checked the impact of introducing other methods of integrating answers of base classifiers. We created the best variant of our approach, i.e. $Bag_{16}DFS_3$ extended by using either Dynamic Voting method or Stacked Combiner (learned also by C4.8 algorithm) for integration of base classifier answers instead of Equal Weight Voting. We also used it for the bagging with only random feature selection iterations denoted as $BagFs + DV$. The results are given in table 3.

As the variant $Bag_{16}DFS_3 + DV$ led to the best improvement of classification accuracy, we checked the possibility of introducing the wrapper approach inside it. It was used as a new feature selection method instead of using the Contextual Merit. Classification results for it, denoted as $BagDFS + wrap$ are presented in table 4. It should be remarked that it significantly improved the classification accuracy (mainly for *ecoli* and *glass* data sets). On the other hand, it also increased computational costs.

Table 4. $BagFs$ vs. $BagDFS$ comparison.

Data set	$BagFs$	$BagDFS$	$BagDFS$ +wrap
glass	77.01 ±1.80	76.87 ±1.87	77.43 ±1.82
bupa	71.91 ±1.81	71.16 ±1.0	71.97 ±1.91
vote	94.83 ±0.39	95.00 ±0.11	94.97 ±0.1
breast	94.56 ±0.75	96.18 ±0.22	96.39 ±0.26
election	92.23 ±0.66	92.50 ±0.53	92.66 ±0.44
wine	95.00 ±1.42	97.08 ±1.02	97.36 ±0.71
ecoli	84.61 ±0.74	83.86 ±0.91	85.39 ±1.02
german	73.65 ±1.12	74.79 ±0.61	74.80 ±0.96

7 Conclusions

In this paper we discussed the ensemble approaches to classification that have attracted a great deal of interest in recent years. These approaches can outperform single classifiers on wide range of classification problems (in particular complex ones). We started from the brief discussion of basic approaches that manipulate input data to obtain diversity of component sub-classifiers inside an ensemble. Our original contribution is proposing an extension of the bagging classifier, by introducing into its structure several different feature selection methods. Moreover, we proposed the usage of new methods for integrating answers of these sub-classifiers, in particular a dynamic voting instead of simple voting combination rule. According to our best knowledge it is an original proposal, which have not been studied yet.

All the extensions were evaluated in the comprehensive experiments. Let us summarize the main results.

- The first observation is that all versions of the extended bagging approach are competitive comparing to the standard version of the bagging classifier. However, one should notice that not for all data these approaches are superior – Bag_{49} is still the best for *bupa* data set, which is a quite difficult imbalanced medical data set.
- The best version of the bagging classifier proposed in this paper, called $Bag_{16}DFS_3 + DV$, is significantly better (in the sense of t-Student paired statistical test) than the previously known bagging variant with random multiple feature selection (Bag_7Fs_7) on 3 out of 8 data sets (precisely *breast, wine* and *german*) The proposed solution consisted of 16 bootstrap samples duplicated 3 times, each time with use of a different feature selection method (i.e. Correlation-based measure, Contextual Merit measure and Plain Random drawing). Introducing the wrapper method instead of the Contextual Merit measure slightly increased the classification accuracy for the extended bagging. One should also notice that some of these data sets with insignificant difference were the smallest sets in terms of a number of examples, while the new $Bag_{16}DFS_3 + DV$ was generally better with increasing a number of examples.
- All extended bagging classifiers *BagDFS* and *BagFs* are significantly better than a single C4.8 classifier.
- Implementing dynamic voting to combine answers of base classifiers led to slightly better results for the $Bag_{16}DFS_3{+}DV$ classifier, while having rather less influence on the *BagFs* classifier. However, no progress was noticed for incorporating *Stacked Combiner* - perhaps other meta-learning algorithm should be chosen.
- Using unprunned trees instead of prunned ones for *bagging* led to accuracy improvement, which fact is consistent with observation made by other researchers [2].

Although the proposed extended classifier is more accurate, one should also take into account the growth of computational costs in comparison to the traditional approach. In our experiments single C4.8 classifier was built on the *glass* data set in 1.5 second, Bag_{49} in 27 seconds, Bag_7Fs_7 in 26 seconds and $Bag_{16}DFS_3$ in 234 seconds. Thus, if the time restrictions are important, the simple random feature selection could be an acceptable alternative. However, we think that integrating feature selection with the bagging may be an effective solution for some complex learning problems. Different methods of feature selection can be more accurate depending on the characteristics of the analysed data.

Acknowledgment. The author would like thank Michal Kaczmarek, who has worked on the software implementation of the proposed classifier and took part is some experiments while working on his M.Sc. Thesis.

References

1. C. Blake, E. Koegh, C.J. Mertz, Repository of Machine Learning, University of California at Irvine (1999).
2. L. Breiman, Bagging predictors. *Machine Learning*, 24 (2), 1996, 123–140.

3. M. Dash, H. Liu, Feature selection for classification. *Intelligent Data Analysis*, 1 (3), 1997.
4. T.G. Dietrich, Ensemble methods in machine learning. In: Kittler J., Roli F. (eds), *Proc. of 1st Int. Workshop on Multiple Classifier Systems*, Springer Verlag LNCS 1857, 2000, 1–15.
5. M. Hall, Correlation-based feature selection for discrete and numeric class machine learning. In: *Proc. 17th Conf. on Machine Learning*, 2000.
6. T. Hastie, R. Tibshirani, J. H. Friedman, *The Elements of Statistical Learning.* Spinger-Verlag, 2001.
7. T.K. Ho, The random subspace method for constructing decision forests. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 20 (8), 1998, 832-844.
8. S.J. Hong, Use of contextual information for feature ranking and discretization. IEEE *Transactions on Knowledge and Data Engineering*, 9, 1997, 718-730.
9. J. Gama, Combining classification algorithms. Ph.D. Thesis, University of Porto, 1999.
10. M. Kaczmarek, *Feature Selection and Multiple Classifiers*, M.Sc. Thesis Poznan Univerity of Technology, 2002.
11. R. Kohavi, D. Sommerfield, Feature Subset Selection Using the Wrapper Method: Overfitting and Dynamic Search Space Topology. *Proceedings of the 1st Int. Conference on Knowledge Discovery and Data Mining*, Montreal, AAAI Press, 1995, 192-197.
12. L. Kuncheva, A theoretical study in six classifier fusion strategies. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24 (2), 2002, 281-286.
13. L. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms.* Wiley, 2004.
14. L. Kuncheva, C. Whitaker, Feature Subsets for Classifier Combination. An Enumerative Experiment. In: *Multiple Classifier Systems. Proc. of the 2nd Int. Workshop* MSC2001, Springer Verlag LNCS 2096, 2001, 228-237.
15. P. Latinne, O. Debeir, Ch. Decaestecker, Mixing Bagging and Multiple Feature Subsets to Improve Classification Accuracy of Decision Tree Combination. *Proc. of the 10th Belgian-Dutch Conf. on Machine Learning*, Tilburg University, 2000.
16. P. Latinne, O. Debeir, Ch. Decaestecker., 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.
17. H. Liu, H. Motoda, *Feature Selection for Data Mining and Knowledge Discovery.* Kluwer Publisher, 1998.
18. R. Maclin, D. Optiz, An empirical evaluation of bagging and boosting. In: *Proceedings of the 14th National Conference on Artificial Intelligence*, Providence, AAAI/MIT Press, 1997, 546-551.
19. R.S. Michalski, I. Bratko, M. Kubat (eds.), *Machine learning and data mining*, John Wiley & Sons, 1998.
20. T. Mitchell, *Machine Learning*, Mac-Graw Hill, Boston, 1997.

21. D. Optiz, Feature selection for ensembles. In: *Proc. of the 16th National Conference on Artificial Intelligence*, AAAI/MIT Press, 1999, 379-384.
22. S. Puuronen, I. Skrypnyk, A. Tsymbal: Ensemble Feature Selection based on Contextual Merit and Correlation Heuristics, In: Proc. of the 5th Conference on Advances in data bases and information systems ADBIS 2001, LNCS 2151, Springer Verlag, 2001, 155 - 168.
23. J. R. Quinlan, *C4.5: Programs for Machine Learning.* Morgan Kaufmann, San Mateo CA, 1993.
24. J.R. Quinlan, *Bagging, boosting and C4.5.* In: Proceedings of the 13th National Conference on Artificial Intelligence, 1996, 725–730.
25. M. Skurichina, R. Duin., Bagging and the random subspace method. In: *Proc. of the Int. Workshop on Multiple Classifier Systems* MCS 2001, LNCS 2096, Springer Verlag, 2001, 1-10.
26. J. Stefanowski, Multiple and hybrid classifiers. In: Polkowski L. (ed.) *Formal Methods and Intelligent Techniques in Control, Decision Making, Multimedia and Robotics*, Post-Proceedings of 2nd Int. Conference, Warszawa, 2001, 174–188.
27. J. Stefanowski, Bagging and induction of decision rules. In: *Post-Proceedings of the Int. Symposium on Intelligent Systems* IIS'2002. Series: Advances of Soft Computing, Physica Verlag, Heidelberg, 2002, 121-130.
28. J. Stefanowski, M. Kaczmarek, Integrating attribute selection and dynamic voting of sub-classifiers to improve accuracy of bagging classifiers. In: *Proc. of the AI-METH 2004. Recent Developments in Artificial Intelligence Methods*, Gliwice, 2004, 263-268.
29. A. Tsymbal, S. Puuronen, Bagging and Boosting with Dynamic Integration of Classifiers. In: *Proceedings of the PKDD'2000 Conference*, Springer Verlag vol. 1910, 2000, 116-125.
30. A. Tsymbal, S. Puuronen, I. Sktypnyk, Ensemble feature selection with dynamic integration of classifiers. In: *Proc. of Int. ICSC Congress on Computational Intelligence Methods and Applications*, CIMA'2001, Bangor, 2001, 558-564.
31. A. Tsymbal, M. Pechenizkiy, P. Cunningham, Diversity in search strategies for ensemble feature selection. *Information Fussion*, 6, 2005, 83-98.
32. G. Valentini, F. Masuli, Ensambles of Learning Machines. In: R. Tagliaferri, M. Marinaro (eds), *Neural Nets WIRN Vietri*-2002, Springer-Verlag LNCS, vol. 2486, 2002 , 3-19.
33. Weka, machine learning software in Java, University of Waikato, http://www.cs.waikato.ac.nz/ml/weka/index.html