Improving Bagging by Feature Selection with Dynamic Integration of Sub-classifiers

Jerzy Stefanowski

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

Abstract. Some applications of machine learning algorithms require improving classification performance. It can be achieved by multiple classifiers which include sets of sub-classifiers, whose individual predictions are combined to classify new objects. These approaches can outperform single classifiers on wide range of classification problems. In this paper we proposed an extension of the popular bagging classifier integrating it with feature subset selection. Moreover, we examined the usage of various methods for aggregating answers of these sub-classifiers, in particular a dynamic voting instead of simple voting combination rule. The experimental results showed that the extended bagging classifier (with decision trees as sub-classifiers) is more accurate than the standard approach.
Keywords: machine learning, data mining, multiple classifiers, bagging, feature selection.

1 Introduction

Machine learning and data mining are domains intensively developed in last years, see e.g. [13]. One of their main sub-domain is *supervised classification learning*, which includes discovering from data knowledge about assigning objects, described by a fixed set of attributes, to one of pre-defined class. The knowledge representation and the strategy of its usage constitute the *classifier*, which can be further applied to predict classes of new or testing objects. A number of various algorithms to discover such knowledge have been introduced and there are several successful applications in industry and other domains - the reader can consult e.g. [13], in particular the survey by Langley and Simon.
Nowadays the most active research in supervised learning includes an *integration* of several base classifiers into the combined classification system [5, 10, 21]. Such systems are known under the names *multiple classifiers, ensembles methods, committees* or *classifier fusion.* This topic attracts an interest of 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., [5, 10, 21]. The most popular 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 sub-classifiers is treated as a necessary condition for their efficient combination [5, 21]. In this paper we will consider only these diversification methods that manipulate input data: either by sampling of learning examples or by feature selection. We will focus our attention on the *bagging* [3], which uses bootstrap sampling to choose different subsets of examples, while selected feature subsets for ensembles could be obtained in many ways (see section 3).
However, the literature study shows that these two diversification dimensions are considered independently. The open research question is - whether integrating both techniques of changing input learning data, i.e. bootstrap sampling and selection of multiple feature subsets, could additionally improve the classification accuracy. According to our best knowledge the most related work is the study by Lattine *et al.* [12], where features were just randomly selected several times over bootstrap samples. Their experimental results showed that this approach performed better on some data sets than the standard bagging. In our paper, we would like to consider enhancing this proposal by integrating bootstrap sampling 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. Thus, the main aim of this paper is to experimentally verify the usefulness of our proposed enhancements of the bagging on its classification performance.
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. [19, 20], it is also important to have a good integration method that utilize the diversity of component sub-classifiers. If some sub-classifiers are more accurate in some subspaces 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 [19]. 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 only [18].
The evaluation is based on many comparative experiments performed on a diverse collection of machine learning benchmark data sets [2]. In all experiments the sub-classifiers are *decision trees* induced by the Ross J. Quinlan $C4.8$ algorithm.

2 Combining Base Classifiers by Bagging

Bagging, introduced by Breiman [3], 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 [3] each bootstrap sample contains 63.2% unique examples from the training set. A family of bootstrap samples $(T_1, \ldots, T_s)$ from the original learning set 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.

Experimental results showed a significant improvement of the classification accuracy [1, 3, 5, 17]. 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. For more theoretical discussion on the bagging the reader is referred to [3].

3 Ensemble Feature Selection - Related Works

The feature subset selection is an important problem in machine learning, or statistics [4, 9, 13]. Typically, this problem is referred to a single learning algorithm and the aim is to find the subset of features leading to not worse classification accuracy than the set of all features. The selection of the attribute subset is 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 [9].

However, 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 are obtained by using different subsets of feature for each of them. Such a problem is also known under the name *ensemble feature selection* [14]. One can have a look to [21, 10] for a review of these approaches. Quite well known is *Random Subspace Method* [7], which consists of training a given number of classifiers, with each having as an input a given proportion k of attributes picked randomly from the original set of attributes. There are also other approaches, where the correlation between each attribute and the output of the class is computed and the base classifier is trained only on the most correlated subset of attributes. Other attribute subspace methods partition the set of attributes in such a way that each subset is used by one classifier.

The discussion and experimental study of partitioning the feature space using different combination schemes led to conclusions that there is no one best feature combination for all situations [10].

4 Aggregating Answers of Sub-classifiers

Another step in creating a multiple classifier is to aggregate the predictions of the base sub-classifiers. In general, there are two kinds of methods: *group combination* or *specialized selection* [10, 20]. 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 sub-classifiers. In its simplest version, 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. The vote of each classifier may be *weighted*, e.g., by the estimating its accuracy. There are also more advanced aggregation rules, e.g. using Bayesian rule or fuzzy aggregation operations - see [21, 10].

Yet 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 [19].

A number of specialized *selection* methods have also been proposed, for review see e.g. [19, 20]. In a case of bagging or feature ensembles the *dynamic integration* methods are often used. In [20] techniques called *Dynamic Selection, Dynamic Voting* were considered. All these are based on local accuracy estimates. 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 [19] 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 [20].

5 Integrating Bagging, Feature Selection and Dynamic Selection of Classifiers

The previously discussed approaches attempt to obtain sub-classifiers diversified either by example sampling or by feature subset selection.

However, both these diversification techniques are considered independently. Here, we will present our concept of joining them.
First s bootstrap samples T_i of the learning set T are generated (with the same sampling schema as in [3]). Then, for each sample we additionally independently select R subsets of features. Finally, $S \cdot R$ new learning sets are obtained to which the same learning algorithm is applied.
In our enhanced approach we would like to select features according to more complex methods than plain random choice only. Let is remind that such a simple random selection has been considered in the most related work [11], where each set contained k proportion of initial features (details of tuning this value - the same for all subsets are given in [11]). The experimental results with this approach $BagFs$, built with decision trees, are encouraging. In our approach we choose more advanced methods evaluating in the different way the relationship between each feature or feature subsets and the decision class. More precisely we replace the R random feature selection iterations for each bootstrap sample by new 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 [18, 17] 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. Due to the size of this paper we skip the detailed results and summarize that finally we chose 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 classes have different values.
- *Info-Gain* : The well known measure based on the information entropy often used in symbolic induction.
- *Chi-Squared statistic*: It is based on widely used statistics to evaluate pairs of features. Any numeric feature have to be discretized [22].
- *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 [6].

As the last method we considered first the Random Subspace Method [7]. In the last phase of experiments we will also use the *wrapper approach* [9], 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 features and the choice of features is done according to their ranking - which requires the parameter k for the best features (details of tuning it are given in [18]).
Additionally, we consider different integration methods to aggregate the answers of sub-classifiers in the proposed enhanced bagging. As we want to compare the usefulness of various 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.

In dynamic voting we compute nearest learning examples of the classified object with an Euclidean distance measure for numeric features and Cost-Salzberg 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 [22] - to be consistent with earlier works [11, 18]. We used standard options of this algorithm with pruning of trees. 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 *election* are coming from UCI repository [2].

First, we chose the configuration, i.e. the number of bootstrap samples. Let is summarize our and others previous experiments [11, 17, 18] with *Bagging* and *BagFs*, which showed that a high number (i.e. up to 343) did not led to much better accuracy while the learning time tended to be too long. The final choice was to consider 49 component classifiers (it is also consistent with the configuration from [11]). So, our standard bagging classifier was built with 49 bootstraps (will be denoted Bag_{49}). Moreover we created a classifier $BagFs$ in the same way as proposed in [11] - it resulted from 7-bootstrap-7-random-chosen-feature-subsets. Then, we constructed an ensemble according to our proposal described in the previous section. As we wanted to construct a multiple classifier as similar in a structure as previous ones 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 $Bag_{10}DFS_5$. In all configurations the equal weight voting was used as an aggregation method. Results of comparison their classification accuracy are given in Table 1. The second column contains performance of the standard single decision tree induced by C4.5.

One can notice that the single classifier is always outperformed by bagging approaches. Differences between versions of bagging depend on data sets. As comparing Bag_7Fs_7 with $Bag_{10}DFS_5$ we observed quite similar performance, we additionally decided to check other versions of our $Bag-DFS$ approach. So, we studied new configuration using only 4 feature selection iterations for each bootstrap (in this case the classifier used

Table 1. Comparison of classification accuracy for standard classifier and bagging (an average value with a standard deviation represented in %).

Data	C4.8	Bag_{49}	Bag_7Fs_7	$Bag_{10}DFS_5$
glass	67.76±1.44	74.77±1.62	77.01±1.80	76.87±2.2
bupa	65.42±1.21	73.62±0.85	71.91±1.81	70.32±1.64
vote	94.23±0.65	94.80±0.28	94.83±0.39	94.97±0.11
breast	94.48±0.62	96.25±0.39	94.56±0.75	95.99±0.38
election	90.56±0.66	91.22±0.76	92.23±0.66	91.73±0.48
wine	93.82±1.18	96.07±0.88	95.00±1.44	96.69±1.04
ecoli	83.10±1.04	84.38±0.80	84.61±0.74	83.99±1.2
german	69.22±1.30	74.14±0.88	73.65±1.12	74.43 ±0.98

12 bootstraps to be similar to total number of 49 components). These configurations 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. We also check next configurations with 3 feature selection iterations for each of 16 bootstraps. We skip detailed results and remark that the most accurate variant is $BagDFS$ without considering Info-Gain and Chi-Squared statistics – its performance is given in Table 2 in column 2 as $Bag_{16}DFS_3 + EV$.

Table 2. Equal weight voting, stacked combination vs. dynamic voting comparison.

Data set	$Bag_{16}DFS_3$ $+EV$	$Bag_{16}DFS_3$ $+DV$	$Bag_{16}DFS_3$ $+SC$	$BagFS$ $+DV$
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 next experiments we checked the impact of introducing other aggregation methods. 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 2.
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 (we skipped it before because its high computational

costs [17]). It was used as a new feature selection method instead of using the weakest Contextual Merit. Classification results for it, denoted as $BagDFS + wrap$ are presented in final Table 3.

Table 3. Classification accuracy for different bagging configurations.

Data set	Bag_{49}	Bag_7Fs_7	$BagDFS$	$BagDFS$ + wrapper
glass	74.77±1.62	77.01±1.80	76.87±1.87	77.43±1.82
bupa	73.62±0.85	71.91±1.81	71.16±1.0	72.46±1.9
vote	94.80±0.28	94.83±0.39	95.00±0.11	94.97±0.1
breast	96.25±0.39	94.56±0.75	96.18±0.22	96.39±0.26
election	91.22±0.76	92.23±0.66	92.50±0.53	92.73±0.48
wine	96.07±0.88	95.00±1.44	97.08±1.02	97.36±0.71
ecoli	84.38±0.80	84.61±0.74	83.86±0.91	85.39±1.02
german	74.14±0.88	73.65±1.12	74.79±0.61	74.86±0.96

7 Discussion of Experimental Results and Final Remarks

The original contribution of this paper 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. All the extensions were evaluated in the comprehensive experiments. Let us discuss their 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*) while for other data improvements are not significant. 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 $Bag_{16}DFS_3$ 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 slight accuracy improvement, which fact is consistent with observation made by other researchers [3].

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. For instance 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.

References

1. E. Bauer, R. Kohavi, An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. Machine Learning, 36 (1/2), 1999, 105–139.
2. C. Blake, E. Koegh, C.J. Mertz, Repository of Machine Learning, University of California at Irvine (1999).
3. L. Breiman, Bagging predictors. *Machine Learning*, 24 (2), 1996, 123–140.
4. M. Dash, H. Liu, Feature selection for classification. *Intelligent Data Analysis*, 1 (3), 1997.
5. 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.
6. M. Hall, Correlation-based feature selection for discrete and numeric class machine learning. In: *Proc. 17th Conf. on Machine Learning*, 2000.
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. 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.

10. L. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*. Wiley, 2004.
11. 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.
12. 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.
13. R.S. Michalski, I. Bratko, M. Kubat (eds.), *Machine learning and data mining*, John Wiley & Sons, 1998.
14. D. Optiz, Feature selection for ensembles. In: *Proc. of the 16th National Conference on Artificial Intelligence*, AAAI/MIT Press, 1999, 379-384.
15. 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.
16. J. R. Quinlan, *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo CA, 1993.
17. J. Stefanowski, An experimental evaluation of improving rule based classifiers with two approaches that change representations of learning examples. *Engineering Applications of Artificial Intelligence Journal*, vol. 17, 2004, 439-445.
18. J. Stefanowski, M. Kaczmarek, Integrating attribute selection to improve accuracy of bagging classifiers. In: *Proc. of the AI-METH 2004. Recent Developments in Artificial Intelligence Methods*, Gliwice, 2004, 263-268.
19. 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.
20. 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.
21. 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.
22. Weka, machine learning software in Java, University of Waikato, http://www.cs.waikato.ac.nz/ml/weka/index.html