World Academy of Science, Engineering and Technology
International Journal of Mathematical and Computational Sciences
Vol:1, No:8, 2007

Combining Bagging and Boosting

S. B. Kotsiantis, P. E. Pintelas

Open Science Index, Mathematical and Computational Sciences Vol:1, No:8, 2007 publications.waset.org/4742/pdf

***Abstract*—** Bagging and boosting are among the most popular re-sampling ensemble methods that generate and combine a diversity of classifiers using the same learning algorithm for the base-classifiers. Boosting algorithms are considered stronger than bagging on noise-free data. However, there are strong empirical indications that bagging is much more robust than boosting in noisy settings. For this reason, in this work we built an ensemble using a voting methodology of bagging and boosting ensembles with 10 sub-classifiers in each one. We performed a comparison with simple bagging and boosting ensembles with 25 sub-classifiers, as well as other well known combining methods, on standard benchmark datasets and the proposed technique was the most accurate.

***Keywords*—** data mining, machine learning, pattern recognition.

I. Introduction

Both empirical observations and specific machine learning applications confirm that a given learning algorithm outperforms all others for a specific problem or for a specific subset of the input data, but it is unusual to find a single expert achieving the best results on the overall problem domain [1]. As a consequence multiple learner systems (an ensemble of classifiers) try to exploit the local different behavior of the base learners to enhance the accuracy and the reliability of the overall inductive learning system. There are also hopes that if some learner fails, the overall system can recover the error.

Numerous methods have been suggested for the creation of ensemble of classifiers [2]. Mechanisms that are used to build ensemble of classifiers include: i) Using different subset of training data with a single learning method, ii) Using different training parameters with a single training method, iii) Using different learning methods.

An accessible and informal reasoning, from statistical, computational and representational viewpoints, of why ensembles can improve results is provided in [2]. The key for success of ensembles is whether the classifiers in a system are diverse enough from each other, or in other words, that the individual classifiers have a minimum of failures in common. If one classifier makes a mistake then the others should not be likely to make the same mistake.

Two of the most popular ensemble algorithms are bagging [3] and boosting [4]. There are two major differences between bagging and boosting. Firstly, boosting changes adaptively the distribution of the training set based on the performance of previously created classifiers while bagging changes the distribution of the training set stochastically. Secondly, boosting uses a function of the performance of a classifier as a weight for voting, while bagging uses equal weight voting. Boosting algorithms are considered stronger than bagging on noise-free data; however, bagging is much more robust than boosting in noisy settings. For this reason, in this work, we built an ensemble combing bagging and boosting version of the same learning algorithm using the sum voting methodology. We performed a comparison with simple bagging and boosting ensembles as well as other known ensembles on standard benchmark datasets and the proposed technique had the best accuracy in most cases. For the experiments, representative algorithms of well known machine learning techniques, such as decision trees, rule learners and Bayesian classifiers were used.

Section 2 presents the most well known methods for building ensembles that are based on a single learning algorithm, while section 3 discusses the proposed ensemble method. Experiment results using a number data sets and comparisons of the presented combining method, using different base classifiers, with other ensembles are presented in section 4. We conclude in Section 5 with summary and further research topics.

II. ENSEMBLES OF CLASSIFIERS

Learning algorithms try to find a hypothesis in a given space *H* of hypotheses and in many cases if we have sufficient data they can find the optimal one for a given problem. But in real cases we have only limited data sets and sometimes only few examples are available. In these cases the learning algorithm can find different hypotheses that appear equally accurate with respect to the available training data, and although we can sometimes select among them the simplest or the one with the lowest capacity, we can avoid the problem combining them to get a good approximation of the unknown true hypothesis.

Thus, there is a growing realization that combinations of classifiers can be more effective than single classifiers. Why rely on the best single classifier, when a more reliable and accurate result can be obtained from a combination of several? This essentially is the reasoning behind the idea of multiple classifier systems.

This section provides a brief survey of methods for constructing ensembles using a single learning algorithm. This

S. B. Kotsiantis is with the University of Patras, P.O. Box: 1399, Greece (phone: +30-2610-997833; fax: +30-2610-997313; e-mail: sotos@ math.upatras.gr).

P. E. Pintelas is with the University of Patras, P.O. Box: 1399, Greece (e-mail: pintelas@ math.upatras.gr).

Open Science Index, Mathematical and Computational Sciences Vol:1, No:8, 2007 publications.waset.org/4742/pdf

set of ensemble creating techniques relies on varying the data in some way. Methods of varying the data include; sampling, use of different data sources, use of different pre-processing methods, distortion, and adaptive re-sampling.

Probably the most well-known sampling approach is that exemplified by bagging [3]. Given a training set, bagging generates multiple bootstrapped training sets and calls the base model learning algorithm with each of them to yield a set of base models. Given a training set of size *t*, bootstrapping generates a new training set by repeatedly (*t* times) selecting one of the *t* examples at random, where all of them have equal probability of being selected. Some training examples may not be selected at all and others may be selected multiple times. A bagged ensemble classifies a new example by having each of its base models classify the example and returning the class that receives the maximum number of votes. The hope is that the base models generated from the different bootstrapped training sets disagree often enough that the ensemble performs better than the base models.

Breiman [3] made the important observation that instability (responsiveness to changes in the training data) is a prerequisite for bagging to be effective. A committee of classifiers that all agree in all circumstances will give identical performance to any of its members in isolation.

If there is too little data, the gains achieved via a bagged ensemble cannot compensate for the decrease in accuracy of individual models, each of which now sees an even smaller training set. On the other end, if the data set is extremely large and computation time is not an issue, even a single flexible classifier can be quite adequate.

Another method that uses different subsets of training data with a single learning method is the boosting approach [4]. It assigns weights to the training instances, and these weight values are changed depending upon how well the associated training instance is learned by the classifier; the weights for misclassified instances are increased. Thus, re-sampling occurs based on how well the training samples are classified by the previous model. Since the training set for one model depends on the previous model, boosting requires sequential runs and thus is not readily adapted to a parallel environment. After several cycles, the prediction is performed by taking a weighted vote of the predictions of each classifier, with the weights being proportional to each classifier's accuracy on its training set.

AdaBoost is a practical version of the boosting approach [4]. There are two ways that Adaboost can use these weights to construct a new training set to give to the base learning algorithm. In boosting by sampling, examples are drawn with replacement with probability proportional to their weights. The second method, boosting by weighting, can be used with base learning algorithms that can accept a weighted training set directly. With such algorithms, the entire training set (with associated weights) is given to the base-learning algorithm.

MultiBoosting [5] is another method of the same category that can be considered as wagging committees formed by AdaBoost. Wagging is a variant of bagging; bagging uses re-sampling to get the datasets for training and producing a weak hypothesis, whereas wagging uses re-weighting for each training example, pursuing the effect of bagging in a different way.

Melville and Mooney [6] present a new meta-learner (DECORATE, Diverse Ensemble Creation by Oppositional Re-labeling of Artificial Training Examples) that uses an existing "strong" learner (one that provides high accuracy on the training data) to build a diverse committee. This is accomplished by adding different randomly constructed examples to the training set when building new committee members. These artificially constructed examples are given category labels that disagree with the current decision of the committee, thereby directly increasing diversity when a new classifier is trained on the augmented data and added to the committee.

III. PROPOSED METHODOLOGY

Recently, several authors [3], [7] have proposed theories for the effectiveness of bagging and boosting based on bias plus variance decomposition of classification error. In this decomposition we can view the expected error of a learning algorithm on a particular target function and training set size as having three components:

1. A bias term measuring how close the average classifier produced by the learning algorithm will be to the target function;
2. A variance term measuring how much each of the learning algorithm's guesses will vary with respect to each other (how often they disagree); and
3. A term measuring the minimum classification error associated with the Bayes optimal classifier for the target function (this term is sometimes referred to as the intrinsic target noise).

Unlike bagging, which is largely a variance reduction method, boosting appears to reduce both bias and variance. After a base model is trained, misclassified training examples have their weights increased and correctly classified examples have their weights decreased for the purpose of training the next base model. Clearly, boosting attempts to correct the bias of the most recently constructed base model by focusing more attention on the examples that it misclassified. This ability to reduce bias enables boosting to work especially well with high-bias, low-variance base models.

As mentioned in [7] the main problem with boosting seems to be robustness to noise. This is expected because noisy examples tend to be misclassified, and the weight will increase for these examples. They present several cases were the performance of boosting algorithms degraded compared to the original algorithms. On the contrary, they point out that bagging improves the accuracy in all datasets used in the experimental evaluation.

Bagging uses a voting technique which is unable to take into account the heterogeneity of the instance space. When the

Open Science Index, Mathematical and Computational Sciences Vol:1, No:8, 2007 publications.waset.org/4742/pdf

majority of the base classifiers give a wrong prediction for a new instance then the majority vote will result in a wrong prediction. The problem may consist in discarding base classifiers (by assigning small weights) that are highly accurate in a restricted region of the instance space because this accuracy is swamped by their inaccuracy outside the restricted area. It may also consist in the use of classifiers that are accurate in most of the space but still unnecessarily confuse the whole classification committee in some restricted areas of the space. The advantage of boosting over bagging is that boosting acts directly to reduce error cases, whereas bagging works indirectly.

For additional improvement of the prediction of a classifier, we suggest combing bagging and boosting methodology with sum rule voting (Vote B&B). When the sum rule is used each sub-ensemble has to give a confidence value for each candidate. In our algorithm, voters express the degree of their preference using as confidence score the probabilities of sub-ensemble prediction. Next all confidence values are added for each candidate and the candidate with the highest sum wins the election. The proposed ensemble is schematically presented in Fig. 1, where h_i is the produced hypothesis of each sub-ensemble, x the instance for classification and y* the final prediction of the proposed ensemble.

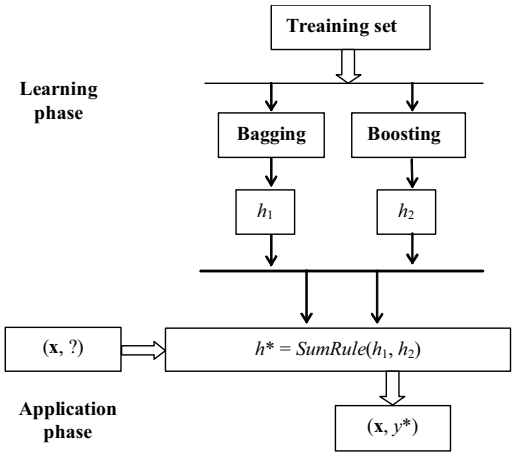


Fig. 1. The proposed ensemble

It has been observed that for bagging and boosting, an increase in committee size (sub-classifiers) usually leads to a decrease in prediction error, but the relative impact of each successive addition to a committee is ever diminishing. Most of the effect of each technique is obtained by the first few committee members [3], [4], [7]. We used 10 sub-classifiers for each sub-ensemble for the proposed algorithm.

The proposed ensemble is effective owing to representational reason. The hypothesis space *h* may not contain the true function *f* (mapping each instance to its real class), but several good approximations. Then, by taking weighted combinations of these approximations, classifiers that lie outside of h may be represented.

It must be also mentioned that the proposed ensemble can be easily parallelized (one machine for each sub-ensemble). This parallel execution of the presented ensemble can reduce the training time in half.

IV. Comparisons and Results

For the comparisons of our study, we used 36 well-known datasets mainly from many domains from the UCI repository [8]. These data sets were hand selected so as to come from real-world problems and to vary in characteristics. Thus, we have used data sets from the domains of: pattern recognition (anneal, iris, mushroom, zoo), image recognition (ionosphere, sonar), medical diagnosis (breast-cancer, breast-w, colic, diabetes, heart-c, heart-h, heart-statlog, hepatitis, lymphotherapy, primary-tumor) commodity trading (autos, credit-g) music composition (waveform), computer games (kr-vs-kp, monk1, monk2), various control applications (balance), language morphological analysis (dimin) [9] and prediction of student dropout (student) [10].

Table I is a brief description of these data sets presenting the number of output classes, the type of the features and the number of examples. In order to calculate the classifiers' accuracy, the whole training set was divided into ten mutually exclusive and equal-sized subsets and for each subset the classifier was trained on the union of all of the other subsets. Then, cross validation was run 10 times for each algorithm and the median value of the 10-cross validations was calculated.

In the following Tables, we represent with "*" that the specific ensemble looses from the base classifier. That is, the specific algorithm performed statistically better than the specific ensemble according to t-test with $p<0.01$. In addition, in Tables, we represent with "v" that the base classifier looses from the specific ensemble according to t-test with $p<0.01$. In all the other cases, there is no significant statistical difference between the results (Draws). It must be mentioned that the conclusions are based on the resulting differences for $p<0.01$ because a p-value of 0.05 is not strict enough, if many classifiers are compared in numerous data sets [11].

In the last rows of the Tables one can see the aggregated results in the form (a/b/c). In this notation "a" means that the specific ensemble algorithm is significantly more accurate than the base algorithm in a out of 36 data sets, "c" means that the base algorithm is significantly more accurate than the specific ensemble in c out of 36 data sets, while in the remaining cases (b), there is no significant statistical difference between the results.

For both Bagging and Boosting, much of the reduction in error appears to have occurred after ten to fifteen classifiers. But Adaboost continues to measurably improve their test-set error until around 25 classifiers [1]. For this reason, we used 25 sub-classifiers for our experiments.

The time complexity of the proposed ensemble is less than both bagging and boosting with 25 sub-classifiers. This happens because we use 10 sub-classifiers for each sub-

ensemble (totally 20). The proposed ensemble also uses less time for training than both Multiboost and Decorare combining methods.

Table I. Description of the data sets

Data sets	Instances	Categ. features	Numer. features	Classes
Anneal	898	32	6	6
audiology	226	69	0	24
autos	205	10	15	6
badge	294	4	7	2
Balance	625	0	4	3
breast-cancer	286	9	0	2
breast-w	699	0	9	2
Colic	368	15	7	2
credit-g	1000	13	7	2
Diabetes	768	0	8	2
dimin	3949	12	0	5
glass	214	0	9	6
haberman	306	0	3	2
heart-c	303	7	6	5
heart-h	294	7	6	5
heart-statlog	270	0	13	2
hepatitis	155	13	6	2
hypothy-roid	3772	22	7	4
ionosphere	351	34	0	2
iris	150	0	4	3
kr-vs-kp	3196	35	0	2
lympho-therapy	148	15	3	4
monk1	124	6	0	2
monk2	169	6	0	2
primary-tumor	339	17	0	21
segment	2310	0	19	7
sick	3772	22	7	2
sonar	208	0	60	2
soybean	683	35	0	19
student	344	11	0	2
titanic	2201	3	0	2
vote	435	16	0	2
vowel	990	3	10	11
waveform	5000	0	40	3
wine	178	0	13	3
Zoo	101	16	1	7

In the following subsection, we present the experiment results for different base classifiers. For the experiments, representative algorithms of well known machine learning techniques, such as decision trees, rule learners and Bayesian classifiers were used. In detail, C4.5 [12], Naïve Bayes [13], OneR [14] and Decision stump [15] are used as base classifier. We have tried to minimize the effect of any expert bias by not attempting to tune any of the algorithms to the specific data set. Wherever possible, default values of learning parameters were used. This naïve approach results in lower estimates of the true error rate, but it is a bias that affects all the learning algorithms equally.

A. Using decision tree as base classifier

Firstly, we used a decision tree algorithm as base classifier in the ensemble. Decision trees are trees that classify examples by sorting them based on attribute values. Each node in a decision tree represents an attribute in an example to be classified, and each branch represents a value that the node can take. Examples are classified starting at the root node and sorting them based on their attribute values. The attribute that best divides the training data would be the root node of the tree. The same process is then repeated on each partition of the divided data, creating sub trees until the training data sets are divided into subsets of the same class. However, a decision tree is said to overfit training data if there exists another hypothesis h´ that has a larger error than h when tested on the training data, but a smaller error than h when tested on the entire data set. For this reason, there are two common approaches that decision tree algorithms can use to avoid overfitting training data: 1) Stop the training algorithm before it reaches a point in which it perfectly fits the training data, 2) Prune the induced decision tree. The most commonly used C4.5 algorithm [12] was the representative of the decision trees in our study. At each level in the partitioning process a statistical property known as information gain is used by C4.5 algorithm to determine which attribute best divides the training examples. The approach that C4.5 algorithm uses to avoid overfitting is by converting the decision tree into a set of rules (one for each path from the root node to a leaf) and then each rule is generalized by removing any of its conditions that will improve the estimated accuracy of the rule.

Decision trees are very unstable in this regard as small perturbations in the training data set can produce large differences in the structure (and predictions) of a model. Bagging and boosting decision trees has been proved to be very successful for many machine-learning problems [3], [16], [7].

Subsequently, we compare the presented ensemble with bagging, boosting and MultiBoost version of C4.5 (using 25 sub-classifiers), as well as, with DECORATE combining method using C4.5 as base classifier. In the last raw of the Table II one can see the concentrated results. The presented ensemble is significantly more accurate than single C4.5 in 7 out of the 36 data sets, while it has significantly higher error rates in none data set. Bagging C4.5 is significantly more accurate than single C4.5 in 3 out of the 36 data sets whilst it has significantly higher error rates in none data set. Furthermore, Adaboost C4.5 has significantly lower error rates in 8 out of the 36 data sets than single C4.5, whereas it is significantly less accurate in 3 data sets. What is more, Multiboost C4.5 is significantly more accurate than single C4.5 in 7 out of the 36 data sets whilst it has significantly higher error rates in one data set. DECORATE C4.5 has significantly lower error rates in 6 out of the 36 data sets than

Open Science Index, Mathematical and Computational Sciences Vol:1, No:8, 2007 publications.waset.org/4742/pdf

single C4.5, whereas it is significantly less accurate in 1 dataset.

Table II. Comparing the proposed ensemble with other well known ensembles that uses as base classifier the C4.5

DataSets	C4.5	Bagging C4.5	Adaboost C4.5	Multiboost C4.5	Decorate C4.5	Vote B&B C4.5
anneal	98.57	98.83	99.61 v	99.59 v	98.75	99.62 v
audiology	77.26	81.29	84.62 v	85.31 v	82.06	84.92 v
autos	81.77	83.89	86.05	85.75	83.86	85.7
badges	100	100	100	100	100	100
balance-scale	77.82	82.33 v	76.91	79.26	81.31 v	79.16
breast-cancer	74.28	73.37	66.5 *	68.27	70.97	69.79
breast-w	95.01	96.31	96.51	96.51	96.5	96.24
colic	85.16	85.23	82.01	83.13	84.82	83.53
credit-g	71.25	74.17	72.79	74.35	72.99	72.63
diabetes	74.49	75.67	72.81	74.21	75.28	73.1
dimin	97.09	97.07	96.03 *	96.17 *	97.2	96.6
glass	67.63	74.86	77.3 v	77.1 v	73	75.85 v
haberman	71.05	72.06	71.12	71.12	73.42	72.16
heart-c	76.94	79.54	79.6	79.87	79.12	79.22
heart-h	80.22	79.91	78.28	80.11	79.3	79.4
heart-statlog	78.15	81.11	80.15	81.37	80.41	80.33
hepatitis	79.22	81.63	82.74	83.59	82.45	82.78
hypothyroid	99.54	99.58	99.67	99.68	98.58 v	99.65
ionosphere	89.74	92.23	93.62	93.54	92.68	93.02
iris	94.73	94.8	94.47	94.47	94.93	94.33
kr-vs-kp	99.44	99.45	99.62	99.62	99.25	99.61
lymphography	75.84	79.14	83.09	83.24	79.09	80.67
monk1	80.61	82.32	96.54 v	94.36 v	89.94	93.87 v
monk2	57.75	61.15	61.86	61.28	58.33	60.97
primary-tumor	41.39	44.4	41.65	41.65	44.57	43.28
segment	96.79	97.54	98.42 v	98.36 v	98.09 v	98.15 v
sick	98.72	98.86	99.06	99.08	98.49 v	99
sonar	73.61	79.03	83.03 v	82.54	82.31	80.08
soybean	91.78	92.78	93.19	93.21	93.86	93.12
students	86.75	86.49	81.44 *	81.68	82.14 *	82.63
titanic	78.55	77.93	78.89	78.71	78.85	78.24
vote	96.57	96.53	95.24	95.51	95.4	96.2
vowel	80.2	91.75 v	95.42 v	94.94 v	96.57 v	93.34 v
waveform	75.25	82.81 v	83.32 v	83.73 v	81.12 v	82.12 v
wine	93.2	95.5	96.62	96.84	97.28	96.56
zoo	92.61	93.29	95.38	95.77	92.96	95.08
W/D/L		*3/33/0*	*8/25/3*	*7/28/1*	*6/29/1*	*7/29/0*

To sum up, the performance of the presented ensemble is more accurate than the other well-known ensembles that use only the C4.5 algorithm (more significant wins than looses in relation to the base algorithm in the used data sets). The proposed ensemble can achieve a reduction in error rate about 15% compared to simple C4.5.

B. Using Decision Stump as base classifier

Secondly, we used decision stump (DS) as base classifier in the ensemble. Decision stumps are one level decision trees [19] that classify instances by sorting them based on feature values [15]. Each node in a decision stump represents a feature in an instance to be classified, and each branch represents a value that the node can take. Instances are classified starting at the root node and sorting them based on their feature values. At worst a decision stump will reproduce the most common sense baseline, and may do better if the selected feature is particularly informative. We compare the presented methodology with bagging, boosting and MultiBoost version of DS (using 25 sub-classifiers). In addition, we compare the presented ensemble with DECORATE combining method using DS as base classifier. In the last raw of the Table III one can see the aggregated

Open Science Index, Mathematical and Computational Sciences Vol:1, No:8, 2007 publications.waset.org/4742/pdf

results. The presented ensemble is significantly more accurate than single DS in 14 out of the 36 data sets, while it has significantly higher error rate in none data set. In addition, the Bagging DS is significantly more accurate than single DS in 5 out of the 36 data sets, whilst it has significantly higher error rate in none data set. Furthermore, Adaboost DS and Decorate DS have significantly lower error rates in 11 and 15 out of the 36 data sets than single DS, respectively whereas they are significantly less accurate in two data sets. Multiboost DS has significantly lower error rates in 10 out of the 36 data sets than single DS, whereas it is significantly less accurate in one data set.

Table III. Comparing the proposed ensemble with other well known ensembles that uses as base classifier the DS

Datasets	DS	Bagging DS	Adaboost DS	Multiboost DS	Decorate DS	VOTE B&B DS
anneal	77.17	82.62 v	83.63 v	83.63 v	76.89	82.81 v
audiology	46.46	46.46	46.46	46.46	46.46	46.46
autos	44.9	44.9	44.9	44.9	51.81	44.9
badges	100	100	100	100	100	100
balance-scale	56.72	68.88 v	71.77 v	71.77 v	81.25 v	72.6 v
breast-cancer	69.27	73.44	71.55	71.76	75.18 v	73.03
breast-w	92.33	92.63	95.28 v	95.05 v	95.78 v	95.12 v
colic	81.52	81.52	82.72	82.9	82.03	82.42
credit-g	70	70	72.6	71.8	69.71	70.65
diabetes	71.8	72.55	75.37	75.19	76.09 v	74.7 v
dimin	59.31	59.31	59.31	59.31	64.75 v	59.31
glass	44.89	44.99	44.89	44.89	53.12 v	44.81
haberman	71.57	72.74	74.06	73.8	71.61	72.95
heart-c	72.93	75.64	83.11 v	83.54 v	72.43	82.45 v
heart-h	81.78	81.37	82.42	81.91	81.78	81.81
heart-statlog	72.3	75.04	81.81 v	82.89 v	81.48 v	81.52 v
hepatitis	77.62	80.72	81.5	82.21	80.02	81.51
hypothyroid	95.39	95.39	92.97 *	92.97 *	95.39	95.38
ionosphere	82.57	82.54	92.34 v	90 v	90.4 v	87.41 v
iris	66.67	70.33	95.07 v	94.73 v	93.93 v	95.4 v
kr-vs-kp	66.05	66.05	95.08 v	93.9 v	90.43 v	94.09 v
lymphography	75.31	74.63	75.44	74.96	72.25 *	75.44
monk1	73.41	73.41	69.79 *	70.37	70.94 *	73.41
monk2	59.58	61.31	53.99	54.19	61.95	56.67
primary-tumor	28.91	28.91	28.91	28.91	29.09	28.91
segment	28.52	56.54 v	28.52	28.52	53.91 v	54.53 v
sick	96.55	96.55	97.07	97.14	96.57	97.11 v
sonar	72.25	73.16	81.06 v	77.58	72.91	75.56
soybean	27.96	27.88	27.96	27.96	41.42 v	27.74
students	87.22	87.22	87.16	86.95	87.1	87.22
titanic	77.6	77.6	77.83	77.62	77.6	77.6
vote	95.63	95.63	96.41	95.63	95.59	95.22
vowel	17.47	23.52 v	17.47	17.47	32.08 v	21.72 v
waveform	56.82	57.41	67.68 v	66.44 v	68.7 v	59.47 v
wine	57.91	85.16 v	91.57 v	91.17 v	96.45 v	88.7 v
zoo	60.43	60.63	60.43	60.43	61.96	60.43
W/D/L		*5/31/0*	*11/23/2*	*10/25/1*	*15/19/2*	*14/22/0*

To sum up, the performance of the presented ensemble is more accurate than the other well-known ensembles that use only the DS algorithm (more significant wins than looses in relation to the base algorithm in the used data sets). The proposed ensemble can achieve a reduction in error rate about 16% compared to simple DS.

C. Using Bayesian algorithm as base classifier

Thirdly, we used a Bayesian method as base classifier in the ensemble. A Bayesian network is a graphical model for probabilistic relationships among a set of attributes. The Bayesian network structure S is a directed acyclic graph (DAG) and the nodes in S are in one-to-one correspondence

Open Science Index, Mathematical and Computational Sciences Vol:1, No:8, 2007 publications.waset.org/4742/pdf

with the attributes. The arcs represent casual influences among the variables while the lack of possible arcs in *S* encodes conditional independencies. Moreover, an attribute (node) is conditionally independent of its non-descendants given its parents. Using a suitable training method, one can induce the structure of the Bayesian Network from a given training set. In spite of the remarkable power of the Bayesian Networks, there is an inherent limitation. This is the computational difficulty of exploring a previously unknown network. Given a problem described by *n* attributes, the number of possible structure hypotheses is more than exponential in *n*. Naive Bayes algorithm was the representative of the Bayesian networks [13]. It is a simple learning that captures the assumption that every attribute is independent from the rest of the attributes, given the state of the class attribute. We compare the presented methodology with bagging, boosting and MultiBoost version of NB (using 25 sub-classifiers), as well as, with DECORATE combining method using NB as base classifier. In the last raw of the Table IV one can see the aggregated results.

Table IV. Comparing the proposed ensemble with other well known ensembles that uses as base classifier the NB

Datasets	NB	Bagging NB	Adaboost NB	Multiboost NB	Decorate NB	VOTE B&B NB
anneal	86.59	86.94	95.2 v	94.13 v	86.59	92.64 v
audiology	72.64	72.1	78.2	80.1 v	72.16	79.22 v
autos	57.41	57.15	57.12	57.12	57.82	57.04
badges	99.66	99.69	99.66	99.66	96.73	99.66
balance-scale	90.53	90.32	92.11	92.29	90.63	91.28
breast-cancer	72.7	72.7	68.57	69.01	72.94	72.45
breast-w	96.07	96.11	95.55	95.58	95.94	96.07
colic	78.7	78.59	77.46	79.28	78.05	80.44
credit-g	75.16	75.08	75.09	74.71	74.7	75.54
diabetes	75.75	75.7	75.88	76.2	75.3	76.03
dimin	92.86	92.77	92.93	93.95	92.86	94.52 v
glass	49.45	50.05	49.63	49.63	49.5	50.14
haberman	75.06	74.86	73.94	73.94	74.83	75.25
heart-c	83.34	83.44	83.14	83.56	83.41	83.54
heart-h	83.95	83.99	84.67	84.8	84.05	84.36
heart-statlog	83.59	83.78	82.3	82.7	83.67	83.74
hepatitis	83.81	83.93	84.23	84.67	82.92	85.89
hypothyroid	95.3	95.47	95.27	95.33	95.3	95.31
ionosphere	82.17	82.36	91.12 v	91.66 v	83.08	88.41 v
iris	95.53	95.73	95.07	95.07	94.87	95.73
kr-vs-kp	87.79	87.77	95.1 v	95.22 v	87.66	93.04 v
lymphography	83.13	83.14	80.67	82.64	82.98	82.98
monk1	73.38	73.35	72.37	72.42	75.65	73.21
monk2	56.83	56.49	56.83	56.83	57.02	56.78
primary-tumor	49.71	49.62	49.71	49.71	49.18	49.32
segment	80.17	80.26	80.17	80.17	80.1	80.31
sick	92.75	92.77	93.7	93.65	92.75	93.17
sonar	67.71	68.05	81.21 v	81.5 v	67.42	74.58 v
soybean	92.94	92.83	92.02	93.15	92.62	93.73
students	85.7	85.59	85.12	85.47	85.09	85.99
titanic	77.85	77.86	77.86	77.88	78.31 v	77.86
vote	90.02	90.05	95.19 v	95.36 v	89.93	92.88 v
vowel	62.9	63.36	81.32 v	79.42 v	62.08	76.54 v
waveform	80.01	80	80.01	80.29	80.4	80.01
wine	97.46	97.52	96.57	96.57	96.51	96.62
zoo	94.97	95.07	97.23	97.23	94.68	97.43
W/D/L		*0/36/0*	*6/30/0*	*7/29/0*	*1/35/0*	*8/28/0*

The presented ensemble is significantly more accurate than single NB in 8 out of the 36 data sets, while it has significantly higher error rate in none data set. Bagging NB can only slightly increase the average accuracy of single NB

Open Science Index, Mathematical and Computational Sciences Vol:1, No:8, 2007 publications.waset.org/4742/pdf

without achieving significantly more accurate results. In addition, Adaboost NB and Multiboost NB are significantly more accurate than single NB in 6 and 7 out of the 36 data sets respectively, whilst they have significantly higher error rate in none data set. DECORATE NB has significantly lower error rates in 1 out of the 36 data sets than single NB, whereas it is significantly less accurate in none data set.

To sum up, the performance of the presented ensemble is more accurate than the other well-known ensembles that use only the NB algorithm (more significant wins than looses in relation to the base algorithm in the used data sets). The proposed ensemble can achieve a reduction in error rate about 9% compared to simple NB.

D. Using Rule learner as base classifier

Fourthly, we used a rule based algorithm as base classifier in the ensemble. In rule induction systems, a decision rule is defined as a sequence of Boolean clauses linked by logical AND operators that together imply membership in a particular class [17]. The general goal is to construct the smallest rule-set that is consistent with the training data. A large number of learned rules is usually a sign that the learning algorithm tries to "remember" the training set, instead of discovering the assumptions that govern it. During classification, the left hand sides of the rules are applied sequentially until one of them evaluates to true, and then the implied class label from the right hand side of the rule is offered as the class prediction.

OneR [14] is a simple classifier that extracts a set of rules based upon a single attribute. OneR shows that it is easy to get reasonable performance on a variety of classification problems by examining only one attribute.

Open Science Index, Mathematical and Computational Sciences Vol:1, No:8, 2007 publications.waset.org/4742/pdf

Table V. Comparing the proposed ensemble with other well known ensembles that uses as base classifier the OneR

Datasets	OneR	Bagging OneR	Adaboost OneR	Multiboost OneR	Decorate OneR	VOTE B&B OneR
anneal	83.63	83.63	85.68	85.06	83.63	83.63
audiology	46.46	46.46	46.46	46.46	46.46	46.46
autos	61.57	64.72	65.97	66.06	70.51 v	65.6
badges	28.55	28.55	28.55	28.55	43.1 v	28.55
balance-scale	57.09	68.28 v	72.84 v	72.84 v	60.94	71.78 v
breast-cancer	66.91	68.81	70.07	70	66.87	69.89
breast-w	92.01	92.95	95.52 v	95.45 v	92.38	94.77 v
colic	81.52	81.52	81.01	80.32	66.13	81.49
credit-g	66.23	68.02	64.31	64.98	71.98	67.53
diabetes	71.71	72.07	69.65	69.95	89.53	73.1
dimin	89.53	89.53	96.39 v	94.67 v	58.76	89.51
glass	57.09	59.69	56.43	56.81	71.97	59.52
haberman	72.79	72	71.3	71.74	72.83	72.4
heart-c	72.53	76.13	73.31	76.24	80.36	78.06 v
heart-h	80.69	80.87	76.75	77.67	74.93	81.31
heart-statlog	71.26	75.85	71.56	75.26	80.94	77.11 v
hepatitis	82.49	82.41	77.56	78.73	78.15	80.92
hypothyroid	96.43	96.54	96.65	96.72	96.38	96.78
ionosphere	82.48	84.98	89.17 v	88.91 v	85.59	87.77 v
iris	93.4	93.73	94	94.53	93.47	94.27
kr-vs-kp	66.91	67.06	94.43 v	93.55 v	66.91	85.04 v
lymphography	74.77	74.7	80.89	79.47	70.66	75.97
monk1	73.41	73.41	71.1	70.44	74.35	73.16
monk2	58.35	58.4	54.71	54.73	57.22	58.46
primary-tumor	27.74	26.76	27.25	27.25	27.88	27.25
segment	64	65.28	84.41 v	80.83 v	68.38 v	71.55 v
sick	96.29	96.34	96.61	96.08	96.24	96.61
sonar	62.36	70.59 v	64.38	65.69	62.44	65.38
soybean	39.75	40.13	40.63	40.63	40.28	40.63
students	87.22	87.22	86.37	87.04	87.04	87.22
titanic	77.6	77.6	77.74	77.64	77.6	77.6
vote	95.63	95.63	96.48	95.91	95.63	95.54
vowel	32.95	35.85	31.33	31.33	40.6 v	31.33
waveform	53.74	55.01	73.76 v	71.28 v	55.68	60.58 v

wine	78.03	79.05	92.08 v	89.7 v	79.15	85.56 v
zoo	42.59	42.59	42.59	42.59	42.59	42.59
W/D/L		*2/34/0*	*8/28/0*	*8/28/0*	*4/32/0*	*9/27/0*

We compare the presented methodology with bagging, boosting and MultiBoost version of OneR (using 25 sub-classifiers). In addition, we compare the presented ensemble with DECORATE combining method using OneR as base classifier.

In the last raw of the Table V one can see the aggregated results. The presented ensemble is significantly more accurate than single OneR in 9 out of the 36 data sets, while it has significantly higher error rate in none data set. In addition, the Bagging OneR is significantly more accurate than single OneR in 2 out of the 36 data sets, whilst it has significantly higher error rate in none data set. Furthermore, Adaboost OneR and Multiboost OneR have significantly lower error rates in 8 out of the 36 data sets than single OneR, whereas they are significantly less accurate in none data set. Decotate OneR has significantly lower error rates in 4 out of the 36 data sets than single OneR, whereas it is significantly less accurate in none data set.

To sum up, the performance of the presented ensemble is more accurate than the other well-known ensembles that use only the OneR algorithm (more significant wins than looses in relation to the base algorithm in the used data sets). The proposed ensemble can achieve a reduction in error rate about 9% compared to simple OneR.

In general for all tested base classifiers the proposed ensemble achieved lower error than either boosting, bagging, multiboost and decorate combining methods when applied to a base learning algorithm and learning tasks for which there is sufficient scope for both bias and variance reduction

V. CONCLUSION

An ensemble of classifiers is a set of classifiers whose individual decisions are combined in some way (typically by weighted or unweighted voting) to classify new examples. One of the most active areas of research in supervised learning has been to study methods for constructing good ensembles of classifiers. The main discovery is that ensembles are often much more accurate than the individual classifiers that make them up. The main reason is that many learning algorithms apply local optimization techniques, which may get stuck in local optima. For instance, decision trees employ a greedy local optimization approach, and neural networks apply gradient descent techniques to minimize an error function over the training data. As a consequence even if the learning algorithm can in principle find the best hypothesis, we actually may not be able to find it. Building an ensemble may achieve a better approximation, even if no assurance of this is given.

Boosting algorithms are considered stronger than bagging on noise-free data, however, bagging is much more robust than boosting in noisy settings. In this work we built an ensemble using a voting methodology of bagging and boosting ensembles. It was proved after a number of comparisons with other ensembles, that the proposed methodology gives better accuracy in most cases. The proposed ensemble has been demonstrated to (in general) achieve lower error than either boosting or bagging when applied to a base learning algorithm and learning tasks for which there is sufficient scope for both bias and variance reduction. The proposed ensemble can achieve an increase in classification accuracy of the order of 9% to 16% compared to the tested base classifiers.

Our approach answers to some extent such questions as generating uncorrelated classifiers and control the number of classifiers needed to improve accuracy in the ensemble of classifiers. While ensembles provide very accurate classifiers, too many classifiers in an ensemble may limit their practical application. To be feasible and competitive, it is important that the learning algorithms run in reasonable time. In our method, we limit the number of sub-classifiers to 10 in each sub-ensemble.

Finally, there are some open problems in ensemble of classifiers, such as how to understand and interpret the decision made by an ensemble of classifiers because an ensemble provides little insight into how it makes its decision. For learning tasks such as data mining applications where comprehensibility is crucial, voting methods normally result in incomprehensible classifier that cannot be easily understood by end-users. These are the research topics we are currently working on and hope to report our findings in the near future.

REFERENCES

[1] Opitz D. & Maclin R., Popular Ensemble Methods: An Empirical Study, *Artificial Intelligence Research*, Vol. 11, 1999, pp. 169-198.
[2] Dietterich, T.G., Ensemble methods in machine learning. In Kittler, J., Roli, F., eds.: Multiple Classifier Systems. *Lecture Notes Computer Sciences*, Vol. 1857, 2001, pp. 1-15.
[3] Breiman L., Bagging Predictors. *Machine Learning*, Vol. 24, No. 3, 1996, pp. 123-140.
[4] Freund Y. and Robert E. Schapire. Experiments with a New Boosting Algorithm, *Proceedings of ICML'96*, pp. 148-156.
[5] Webb G. I., MultiBoosting: A Technique for Combining Boosting and Wagging, *Machine Learning*, Vol. 40, 2000, pp. 159–196.
[6] Melville P., Mooney R., Constructing Diverse Classifier Ensembles using Artificial Training Examples, *Proceedings of IJCAI-2003*, pp.505-510, Acapulco, Mexico.
[7] Bauer, E. & Kohavi, R., An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, Vol. 36, 1999, pp. 105–139.
[8] Blake, C.L. & Merz, C.J., UCI Repository of machine learning databases. Irvine, CA: University of California, 1998, Department of Information and Computer Science. [http://www.ics.uci.edu/~mlearn/MLRepository.html]
[9] Bosch, A. and Daelemans W., Memory-based morphological analysis. *Proceedings of 37th Annual Meeting of the ACL*, 1999, University of Maryland, pp. 285-292 (http://ilk.kub.nl/~antalb/ltuia/week10.html).
[10] Kotsiantis, S., Pierrakeas, C. and Pintelas, P., Preventing student dropout in distance learning systems using machine learning techniques, *Lecture Notes in AI*, Springer-Verlag Vol 2774, 2003, pp 267-274.
[11] Salzberg, S., On Comparing Classifiers: Pitfalls to Avoid and a Recommended Approach, *Data Mining and Knowledge Discovery*, Vol. 1, 1997, pp. 317–328.

Open Science Index, Mathematical and Computational Sciences Vol:1, No:8, 2007 publications.waset.org/4742/pdf

[12] Quinlan J.R., *C4.5: Programs for machine learning*. 1993, Morgan Kaufmann, San Francisco.
[13] Domingos P. & Pazzani M., On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, Vol. 29, 1997, pp. 103-130.
[14] Holte, R. C., Very simple classification rules perform well on most commonly used datasets, *Machine Learning*, Vol. 11, 1993, pp. 63-90.
[15] Iba, W., & Langley, P., Induction of one-level decision trees, *Proceedings of Ninth International Machine Learning Conference*, 1992. Aberdeen, Scotland.
[16] Schapire, R. E., Freund, Y., Bartlett, P., & Lee, W. S., Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics*, Vol. 26, 1998, pp. 1651–1686.
[17] Furnkranz, J., Separate-and-Conquer Rule Learning, *Artificial Intelligence Review*, Vol. 13, 1999, pp. 3-54.
[18] Jensen F., *An Introduction to Bayesian Networks*. 1996, Springer.
[19] Murthy, Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey, *Data Mining and Knowledge Discovery*, Vol. 2, 1998, pp. 345–389.

Open Science Index, Mathematical and Computational Sciences Vol:1, No:8, 2007 publications.waset.org/4742/pdf

Sotiris Kotsiantis received his Diploma in Mathematics at the Department of Mathematics, University of Patras, Greece in 1999. He received MSc in Computational Mathematics and Informatics from the same Department in 2001. Since then he has been a Phd student in the same Department. His research interests focus on Machine Learning, Data Mining, Student Modeling and Multi-agent Systems. In these fields he has some publications in international scientific journals and conferences.

Dr. Panayotis E. Pintelas received his BSc in Mathematics from the University of Athens, Greece, in 1971, and his MSc and PhD in Computer Science from the University of Bradford, UK, in 1973 and 1976, respectively. Since 1995, he is a Professor of Computer Science with the Informatics Division of the Dept. of Mathematics at the Univ. of Patras. His research interests include Software Specification and Software Design, Software Engineering, Computer Assisted Training and Education (CAL/CAI/CBT/ITS) and Open and Distance Learning; in these fields he has numerous publications in international scientific journals and conferences.