Combining Bagging, Boosting and Dagging for Classification Problems

S.B. Kotsianti and D. Kanellopoulos

Educational Software Development Laboratory
Department of Mathematics
University of Patras, P.A. Box: 1399, Rio 26 500
sotos@math.upatras.gr, dkanellop@teipat.gr

Abstract. Bagging, boosting and dagging are well known 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 and dagging on noise-free data. However, there are strong empirical indications that bagging and dagging are much more robust than boosting in noisy settings. For this reason, in this work we built an ensemble using a voting methodology of bagging, boosting and dagging ensembles with 8 sub-classifiers in each one. We performed a comparison with simple bagging, boosting and dagging ensembles with 25 sub-classifiers, as well as other well known combining methods, on standard benchmark datasets and the proposed technique had better accuracy in most cases.

Keywords: Machine learning, data mining, ensembles of classifiers.

1 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 [5]. 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. Numerous methods have been suggested for the creation of ensemble of classifiers [16]. 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 [5]. 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.

B. Apolloni et al. (Eds.): KES 2007/ WIRN 2007, Part II, LNAI 4693, pp. 493–500, 2007.

Three of the most popular ensemble algorithms are bagging [4], boosting [6] and dagging [14]. 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. On the other hand, dagging creates a number of disjoint, stratified folds out of the data and feeds each chunk of data to a copy of the supplied base learner, while predictions are made via majority vote. Boosting algorithms are considered stronger than bagging and dagging on noise-free data; however, bagging and dagging are much more robust than boosting in noisy settings. For this reason, in this work, we built an ensemble combing bagging, boosting and dagging version of the same learning algorithm using the sum voting methodology. We performed a comparison with simple bagging, boosting and dagging ensembles as well as other known ensembles on standard benchmark datasets and the proposed technique had better accuracy in most cases. For the experiments, decision stump algorithm was used as base learner.

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 proposed combining method with other ensembles are presented in section 4. We conclude with summary and further research topics in Section 5.

2 Ensembles of Classifiers

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 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 [4]. 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 [4] 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 [6]. 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 [6].

Dagging [14] creates a number of disjoint, stratified folds out of the data and feeds each chunk of data to a copy of the supplied base learner. Predictions are made via majority vote. MultiBoosting [15] 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 [9] 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.

3 Proposed Methodology

Recently, several authors [1], [4], [14] have proposed theories for the effectiveness of bagging, boosting and dagging 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.

Unlike bagging and dagging, 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 [1] 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. Bagging uses a voting technique which is unable to take into account the heterogeneity of the instance space. When the 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, boosting and dagging methodology with sum rule voting (Vote B&B&D). 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. It has been observed that for bagging, boosting and dagging, 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 [1], [4], [6]. We used 8 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.

4 Comparisons and Results

For the comparisons of our study, we used 36 well-known datasets mainly from many domains from the UCI repository [2]. 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) [3] and prediction of student dropout (student) [8].

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 average value of the 10-cross validations was calculated [16]. In the following Table, 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 Table, 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 [12]. In the last rows of the Table one can also 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 bagging, boosting and dagging, much of the reduction in error appears to have occurred after ten to fifteen classifiers. But Adaboosting continues to measurably improve their test-set error until around 25 classifiers [11]. For this reason, we used 25 sub-classifiers for our experiments. The time complexity of the proposed ensemble is about the same with simple bagging, boosting and dagging with 25 sub-classifiers. This happens because we use 8 sub-classifiers for each sub-ensemble (totally 24). The proposed ensemble also use less time for training than both Multiboost and Decorare combining methods. In the following subsection, we present the experiment results using Decision stump [7] as base classifier.

4.1 Using Decision Stump as Base Classifier

Decision stumps (DS) are one level decision trees [10] that classify instances by sorting them based on feature values [7]. 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 proposed methodology with bagging, boosting, dagging 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 1 one can see the aggregated results.

Table 1. Comparing the proposed ensemble with other well known ensembles that uses as base classifier the DS

Datasets	DS	Bagging DS	Dagging DS	Adaboost DS	Multiboost DS	Decorate DS	VOTE B&B&D DS
anneal	77.17	82.62 v	83.63 v	83.63 v	83.63 v	76.89	83.43 v
audiology	46.46	46.46	36.34 *	46.46	46.46	46.46	46.46
autos	44.9	44.9	44.52	44.9	44.9	51.81	44.9
badges	100	100	100	100	100	100	100
balance-scale	56.72	68.88 v	81.1 v	71.77 v	71.77 v	81.25 v	75.64 v
breast-cancer	69.27	73.44	72.53	71.55	71.76	75.18 v	73.9 v
breast-w	92.33	92.63	95.05 v	95.28 v	95.05 v	95.78 v	95.14 v
colic	81.52	81.52	79.13	82.72	82.9	82.03	82.14
credit-g	70	70	70.16	72.6	71.8	69.71	70.33
diabetes	71.8	72.55	75.56	75.37	75.19	76.09 v	75.03
dimin	59.31	59.31	59.53	59.31	59.31	64.75 v	59.31
glass	44.89	44.99	47.21 v	44.89	44.89	53.12 v	50.69 v
haberman	71.57	72.74	73.01	74.06	73.8	71.61	72.85
heart-c	72.93	75.64	80.67 v	83.11 v	83.54 v	72.43	82.8 v
heart-h	81.78	81.37	81.71	82.42	81.91	81.78	81.78
heart-statlog	72.3	75.04	81.19 v	81.81 v	82.89 v	81.48 v	81.81 v
hepatitis	77.62	80.72	79.45	81.5	82.21	80.02	80.64
hypothyroid	95.39	95.39	94.97	92.97 *	92.97 *	95.39	95.33
ionosphere	82.57	82.54	81	92.34 v	90 v	90.4 v	86.73 v
iris	66.67	70.33	78.13 v	95.07 v	94.73 v	93.93 v	95.07 v
kr-vs-kp	66.05	66.05	66.94	95.08 v	93.9 v	90.43 v	94.09 v
lymphography	75.31	74.63	73.61	75.44	74.96	72.25 *	75.5
monk1	73.41	73.41	65.68 *	69.79 *	70.37	70.94 *	73.09
monk2	59.58	61.31	55.88	53.99	54.19	61.95	58.28
primary-tumor	28.91	28.91	27.37	28.91	28.91	29.09	29.32
segment	28.52	56.54 v	61.05 v	28.52	28.52	53.91 v	56.66 v
sick	96.55	96.55	96.5	97.07	97.14	96.57	96.5
sonar	72.25	73.16	70.29	81.06 v	77.58	72.91	75.26
soybean	27.96	27.88	43.51 v	27.96	27.96	41.42 v	30.76
students	87.22	87.22	87.16	87.16	86.95	87.1	87.22
titanic	77.6	77.6	77.6	77.83	77.62	77.6	77.6
vote	95.63	95.63	95.61	96.41	95.63	95.59	95.52
vowel	17.47	23.52 v	34.84 v	17.47	17.47	32.08 v	27.75 v
waveform	56.82	57.41	67.5 v	67.68 v	66.44 v	68.7 v	60.45 v
wine	57.91	85.16 v	71.21 v	91.57 v	91.17 v	96.45 v	91.74 v
zoo	60.43	60.63	39.51 *	60.43	60.43	61.96	60.43
W/D/L		*5/31/0*	*12/21/3*	*11/23/2*	*10/25/1*	*15/19/2*	*14/22/0*

The proposed 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. Dagging DS has significantly lower error rates in 12 out of the 36 data sets than single DS, whereas it is significantly less accurate in 3 data sets.

To sum up, the performance of the proposed 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 21% compared to simple DS.

5 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 [13].

Boosting algorithms are considered stronger than bagging and dagging on noise-free data; however, bagging and dagging are much more robust than boosting in noisy settings. In this work we built an ensemble using a voting methodology of bagging, boosting and dagging 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 or dagging 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 21% compared to the tested base classifier.

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 8 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. Bauer, E., Kohavi, R.: An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. Machine Learning 36, 105–139 (1999)
2. Blake, C.L., Merz, C.J.: UCI Repository of machine learning databases. Irvine, CA: University of California, Department of Information and Computer Science (1998), http://www.ics.uci.edu/m learn/MLRepository.html
3. Bosch, A., Daelemans, W.: Memory-based morphological analysis. In: proceedings of 37th Annual Meeting of the ACL, University of Maryland, pp. 285–292 (1999), http://ilk.kub.nl/ãntalb/ltuia/week10.html
4. Breiman, L.: Bagging Predictors. Machine Learning 24(3), 123–140 (1996)
5. Dietterich, T.G.: Ensemble methods in machine learning. In: Kittler, J., Roli, F. (eds.) MCS 2000. LNCS, vol. 1857, pp. 1–15. Springer, Heidelberg (2000)
6. Freund, Y., Schapire, R.E.: Experiments with a New Boosting Algorithm. In: proceedings of ICML'96, pp. 148–156 (1996)
7. Iba, W., Langley, P.: Induction of one-level decision trees. In: proceedings of Ninth International Machine Learning Conference (1992). Aberdeen, Scotland (1992)
8. Kotsiantis, S., Pierrakeas, C., Pintelas, P.: Preventing student dropout in distance learning systems using machine learning techniques. In: Palade, V., Howlett, R.J., Jain, L. (eds.) KES 2003. LNCS, vol. 2774, pp. 267–274. Springer, Heidelberg (2003)
9. Melville, P., Mooney, R.: Constructing Diverse Classifier Ensembles using Artificial Training Examples. In: proceedings of IJCAI-2003, Acapulco, Mexico, pp. 505–510 (August 2003)
10. Murthy: Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey. Data Mining and Knowledge Discovery 2, 345–389 (1998)
11. Opitz, D., Maclin, R.: Popular Ensemble Methods: An Empirical Study. Artificial Intelligence Research 11, 169–198 (1999)
12. Salzberg, S.: On Comparing Classifiers: Pitfalls to Avoid and a Recommended Approach. Data Mining and Knowledge Discovery 1, 317–328 (1997)
13. 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 26, 1651–1686 (1998)
14. Ting, K.M., Witten, I.H.: Stacking Bagged and Dagged Models. In: Fourteenth international Conference on Machine Learning, San Francisco, CA, pp. 367–375 (1997)
15. Webb, G.I.: MultiBoosting: A Technique for Combining Boosting and Wagging. Machine Learning 40, 159–196 (2000)
16. Witten Ian, H., Eibe, F.: Data Mining: Practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)