World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:1, No:1, 2007

Combining Bagging and Additive Regression

Sotiris B. Kotsiantis

Open Science Index, Computer and Information Engineering Vol:1, No:1, 2007 publications.waset.org/6392.pdf

***Abstract*—** Bagging and boosting are among the most popular re-sampling ensemble methods that generate and combine a diversity of regression models using the same learning algorithm as base-learner. 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 an averaging methodology of bagging and boosting ensembles with 10 sub-learners in each one. We performed a comparison with simple bagging and boosting ensembles with 25 sub-learners on standard benchmark datasets and the proposed ensemble gave better accuracy.

***Keywords*—** Regressors, statistical learning.

I. Introduction

In this paper we consider the following regression setting. Data is generated from an unknown distribution P on some domain X and labeled according to an unknown function g. A learning algorithm receives a sample S = {(x_1, $g(x_1)$), . . . , (x_m, $g(x_m)$)} and attempts to return a function f close to g on the domain X. Many regression problems involve an investigation of relationships between attributes in heterogeneous databases, where different prediction models can be more appropriate for different regions.

Both empirical observations and specific statistical 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 [7]. As a consequence multiple learner systems 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 learners [6]. Mechanisms that are used to build ensemble of learners 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.

Two of the most popular ensemble algorithms are bagging [4] and boosting [8]. 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 learners while bagging changes the distribution of the training set stochastically. Secondly, boosting uses a function of the performance of a learner as a weight for averaging, while bagging uses equal weight averaging. 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 averaging methodology. We performed a comparison with simple bagging and boosting ensembles on standard benchmark datasets and we took better accuracy in most cases. For the experiments, representative algorithms of well known statistical learning techniques, such as regression trees and regression rules were used.

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

II. ENSEMBLES OF LEARNERS

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 learners can be more effective than single learners. Why rely on the best single learner, 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 learner systems.

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

Manuscript received January 6, 2007.

Sotiris B. Kotsiantis is with the Educational Software Development Laboratory, Department of Mathematics, University of Patras, P.A. Box: 1399, Rio 26 500, Greece, phone: +30 2610 997833, fax: +30 2610 997313, e-mail: sotos@ math.upatras.gr.

Open Science Index, Computer and Information Engineering Vol:1, No:1, 2007 publications.waset.org/6392.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.

Bagging [4] is a "bootstrap" ensemble method that creates individual regression models by training the same learning algorithm on a random redistribution of the training set. Each regression model's training set is generated by randomly drawing, with replacement, N instances – where N is the size of the original training set. Many of the original instances may be repeated in the resulting training set while others may be left out. After the construction of several regression models, taking the average value of the predictions of each regression model gives the final prediction. A more sophisticated version of bagging is described in [5]

Another method that uses different subset of training data with a single learning method is the boosting approach [8]. The boosting approach uses the base models in sequential collaboration, where each new model concentrates more on the examples where the previous models had high error. Although boosting for regression has not received nearly as much attention as boosting for classification, there is some work examining gradient descent boosting algorithms in the regression context.

The AdaBoost.R algorithm [9] attacks the regression problem by reducing it to a classification problem. Friedman has also explored regression using the gradient descent approach (Additive regression model) [14]. In each iteration Friedman's master algorithm constructs ˜yi -values for each data-point xi equal to the (negative) gradient of the loss of its current master hypothesis on xi. The base learner then finds a function in a class F minimizing the squared error on this constructed sample.

One major issue in combining a set of learned models is the amount of correlation in the set of predictors. A high degree of correlation is expected because the learned models are attempting to perform the same prediction task. Correlation reflects the amount of agreement or linear dependence between models when making a set of predictions. The more the models agree, the more correlation, or redundancy, is present. Such a high degree of correlation in the model set can cause some combining schemes to produce unreliable estimates.

III. PROPOSED METHODOLOGY

Recently, several authors [4], [12] have proposed theories for the effectiveness of bagging and boosting based on bias plus variance decomposition of 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 two components:

- A bias term measuring how close the average learner produced by the learning algorithm will be to the target function;
- A variance term measuring how much each of the learning algorithm's guesses will vary with respect to each other (how often they disagree)

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 [4] 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 an averaging technique which is unable to take into account the heterogeneity of the instance space. When the majority of the base learners give a wrong prediction for a new instance then the averaging decision will result in a wrong prediction. The problem may consist in discarding base learners (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 learners that are accurate in most of the space but still unnecessarily confuse the whole 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 learner, we suggest combing bagging and boosting methodology with averaging (Average B&B). The proposed ensemble is algorithmically presented in Fig. 1 and schematically in Fig. 2, 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.

MODEL GENERATION
Let n be the number of instances in the training data.
For each of t iterations (t=10 in our experiments):

- Sample n instances with replacement from training data.
- Built a learner from the sample
- Store the resulting model.

For each of t iterations (t=10 in our experiments):

- Sample n instances concentrating more on the examples where previous models had high error.
- Built a learner from the sample
- Store the resulting model.

Open Science Index, Computer and Information Engineering Vol:1, No:1, 2007 publications.waset.org/6392.pdf

PREDICTION
For each of the 2t models:
  Predict the value of test instance
Return the average value of the predicted values

Fig. 1 The algorithmically description of the proposed ensemble

It has been observed that for bagging and Additive regression, an increase in committee size (sub-learners) 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 [2], [4], [12]. We used 10 sub-learners for each sub-ensemble for the proposed algorithm.

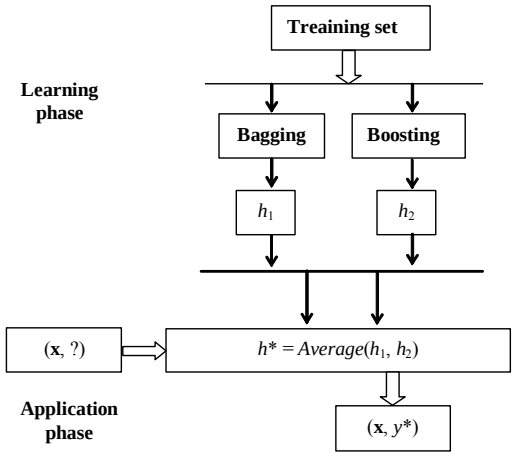

Fig. 2 The schematically description of the proposed ensemble

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, learners 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

We experimented with 29 datasets from the UCI repository [3]. These datasets cover many different types of problems which have discrete, continuous, and symbolic variables. Some datasets have missing values, and some have a mixture of all the above variables. The specific datasets are listed in Table I.

The most well known measure for the degree of fit for a regression model to a dataset is the correlation coefficient. If the actual target values are $a_1, a_2, \ldots a_n$, and the predicted target values are: $p_1, p_2, \ldots p_n$ then the correlation coefficient is given by the formula:

$$R = \frac{S_{PA}}{\sqrt{S_P S_A}} \quad (1) \qquad S_{PA} = \frac{\sum_i (p_i - \bar{p})(a_i - \bar{a})}{n-1} \quad (2)$$

$$S_P = \frac{\sum_i (p_i - \bar{p})^2}{n-1} \quad (3) \qquad S_A = \frac{\sum_i (a_i - \bar{a})^2}{n-1} \quad (4)$$

where, $\bar{p}$: the average value of p_i and $\bar{a}$: the average value of α_i.

TABLE I
THE USED DATASETS

	Instances	Categorical features	Numerical features
auto93	93	6	16
autoHorse	205	8	17
autoMpg	398	3	4
autoPrice	159	0	15
bodyfat	252	0	14
bolts	40	0	7
br.Tumor	286	8	1
cholesterol	303	7	6
cleveland	303	7	6
Cpu	209	1	6
detroit	13	0	13
elusage	55	1	1
fishcatch	158	2	5
gascons	27	0	4
housing	506	1	12
hungarian	294	7	6
longlay	16	0	6
lowbwt	189	7	2
pbc	418	8	10
pharynx	195	11	1
pollution	60	0	15
pwLinear	200	0	10
quake	2178	0	3
sensory	576	11	0
servo	167	4	0
sleep	62	0	7
strike	625	1	5
veteran	137	4	3
vineyard	52	0	3

In order to calculate the regression models' correlation coefficient, the whole training set was divided into ten mutually exclusive and equal-sized subsets and for each subset the regression model 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. It must be mentioned that we used the free available source code for most of the algorithms by [13] for our experiments.

In the following tables, we represent as "v" that the specific algorithm performed statistically better than the proposed ensemble according to t-test with $p<0.05$. Throughout, we speak of two results for a dataset as being "significant different" if the difference is statistical significant at the 5% level according to the corrected resampled t-test [11], with each pair of data points consisting of the estimates obtained in one of the 100 folds for the two learning methods being compared. On the other hand, "*" indicates that proposed

Open Science Index, Computer and Information Engineering Vol:1, No:1, 2007 publications.waset.org/6392.pdf

ensemble performed statistically better than the specific algorithm according to t-test with $p<0.05$. In all the other cases, there is no significant statistical difference between the results (Draws). In the last row of the tables one can also see the aggregated results in the form (α/b/c). In this notation "α" means that the proposed ensemble is significantly less accurate than the compared algorithm in α out of 29 datasets, "c" means that the proposed algorithm is significantly more accurate than the compared algorithm in c out of 29 datasets, while in the remaining cases (b), there is no significant statistical difference.

For both Bagging and Boosting, much of the reduction in error appears to have occurred after ten to fifteen learners. But Additive Regression continues to measurably improve their test-set error until around 25 learners [14]. For this reason, we used 25 sub-learners for our experiments.

The time complexity of the proposed ensemble is less than both bagging and boosting with 25 sub-learners. This happens because we use 10 sub-learners for each sub-ensemble (totally 20).

In the following subsection, we present the experiment results for different base learners. For the experiments, representative algorithms of well known statistical learning techniques, such as regression trees and regression rules were used. In detail, RepTree [13], Decision stump [15] and ConjunctiveRule [13] were used as base learner. 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 RepTree as base learner

Regression trees are binary decision trees with numerical values at the leaf nodes: thus they can represent any piecewise linear approximation to an unknown function. A regression tree is generated in two stages. The first builds an ordinary decision tree, using as splitting criterion the maximization of the intra-subset variation of the target value. The second prunes this tree back by replacing subtrees with a numerical value wherever this seems appropriate.

Regression 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 regression trees has been proved to be very successful for many machine-learning problems [1], [3], [14]. REPTree [13] is a fast regression tree learner that uses information variance reduction and reduced-error pruning (with backfitting).

Subsequently, we compare the presented ensemble with bagging, boosting version of RepTree (using 25 sub-learners). In the last raw of the Table II one can see the concentrated results.

The proposed ensemble is significantly more accurate than single RepTree and Additive Regression RepTree in 27 out of the 29 data sets, while it has significantly lower correlation coefficient in none data set. Furthermore, proposed ensemble has significantly higher correlation coefficient in 4 out of the 29 data sets than Bagging RepTree, whereas it is significantly less accurate in 3 data sets.

To sum up, the performance of the presented ensemble is more accurate than the other well-known ensembles that use only the RepTree algorithm. The proposed ensemble can achieve an increase in correlation coefficient about 20% compared to simple RepTree.

TABLE II
COMPARING THE PROPOSED ENSEMBLE WITH OTHER WELL KNOWN ENSEMBLES THAT USES AS BASE LEARNER THE REPTREE

Datasets	Average B&B RepTree	Bagging RepTree		Additive Regression RepTree		RepTree	
auto93	0.45	0.43	*	0.26	*	0.23	*
autoHorse	0.89	0.89		0.85	*	0.83	*
autoMpg	0.91	0.91		0.89	*	0.88	*
autoPrice	0.92	0.92		0.90	*	0.88	*
bodyfat	0.98	0.98		0.98		0.98	
bolts	0.82	0.84		0.78		0.72	*
breastTumor	0.21	0.22		0.16	*	0.15	*
cholesterol	0.16	0.19	v	0.07	*	0.07	*
cleveland	0.65	0.68	v	0.58	*	0.54	*
cpu	0.95	0.96		0.90	*	0.90	*
detroit	0.20	0.24		0.04	*	0.03	*
elusage	0.82	0.82		0.80	*	0.80	*
fishcatch	0.97	0.97		0.95	*	0.95	*
gascons	0.84	0.82	*	0.71	*	0.67	*
housing	0.90	0.91		0.86	*	0.85	*
hungarian	0.63	0.64		0.58	*	0.58	*
longley	0.52	0.53		0.38	*	0.39	*
lowbwt	0.79	0.79		0.77	*	0.78	
pbc	0.53	0.55		0.46	*	0.46	*
pharynx	0.33	0.01	*	0.3	*	0.01	*
pollution	0.64	0.71	v	0.44	*	0.43	*
pwLinear	0.92	0.92		0.90	*	0.89	*
quake	0.11	0.12		0.06	*	0.07	*
sensory	0.50	0.52		0.46	*	0.45	*
servo	0.92	0.90	*	0.90	*	0.86	*
sleep	0.61	0.63		0.49	*	0.49	*
strike	0.53	0.55		0.40	*	0.40	*
veteran	0.39	0.41		0.24	*	0.23	*
vineyard	0.74	0.74		0.63	*	0.64	*
W-D-L		3/23/3		0/2/27		0/2/27	
Average correlation coefficient	0.65	0.64		0.58		0.56	

B. Using Decision Stump as base learner

Secondly, we used decision stump (DS) as base learner in the ensemble. Decision stump (DS) are one level regression

Open Science Index, Computer and Information Engineering Vol:1, No:1, 2007 publications.waset.org/6392.pdf

trees that classify instances by sorting them based on feature values [10]. 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 and boosting version of DS (using 25 sub-learners).

In the last raw of the Table III one can see the aggregated results. The proposed ensemble is significantly more accurate than single DS in 27 out of the 29 data sets, while it has significantly lower correlation coefficient in none data set. In addition, the proposed ensemble is significantly more accurate than Bagging DS in 23 out of the 29 data sets, whilst it has significantly lower correlation coefficient in one data set. Furthermore, the proposed ensemble has significantly higher correlation coefficient in 10 out of the 29 data sets than Boosting DS, whereas it significantly less accurate in 9 data sets.

TABLE III
COMPARING THE PROPOSED ENSEMBLE WITH OTHER WELL KNOWN ENSEMBLES THAT USES AS BASE LEARNER THE DS

Datasets	Average B&B DS	Bagging DS		Additive Regression DS		DS	
auto93	0.79	0.74	*	0.79		0.59	*
autoHorse	0.86	0.80	*	0.90	v	0.72	*
autoMpg	0.86	0.78	*	0.90	v	0.74	*
autoPrice	0.89	0.82	*	0.91	v	0.81	*
bodyfat	0.93	0.84	*	0.97	v	0.82	*
bolts	0.83	0.71	*	0.81		0.69	*
breastTumor	0.28	0.23	*	0.29		0.22	*
cholesterol	0.15	0.12	*	0.14		0.04	*
cleveland	0.65	0.61	*	0.63	*	0.40	*
cpu	0.95	0.87	*	0.97	v	0.31	*
detroit	0.18	0.23		0.12	*	0.07	*
elusage	0.85	0.84		0.83	*	0.74	*
fishcatch	0.93	0.85	*	0.97	v	0.83	*
gascons	0.90	0.72	*	0.79	*	0.65	*
housing	0.84	0.74	*	0.88	v	0.60	*
hungarian	0.66	0.60	*	0.67		0.56	*
longley	0.45	0.49		0.44		0.33	*
lowbwt	0.79	0.78		0.77	*	0.78	
pbc	0.53	0.46	*	0.53		0.43	*
pharynx	0.70	0.67	*	0.66	*	0.67	*
pollution	0.60	0.60		0.54	*	0.37	*
pwLinear	0.81	0.68	*	0.85	v	0.68	*
quake	0.11	0.09	*	0.08	*	0.09	*
sensory	0.38	0.29	*	0.38		0.29	*
servo	0.84	0.79	*	0.85		0.79	*
sleep	0.53	0.58	v	0.42	*	0.52	
strike	0.44	0.36	*	0.46	v	0.18	*
veteran	0.39	0.33	*	0.39		0.15	*
vineyard	0.70	0.63	*	0.67	*	0.31	*
W-D-L		1/5/23		9/10/10		0/2/27	
Average correlation coefficient	0.65	0.59		0.64		0.5	

To sum up, the performance of the presented ensemble is more accurate than the other well-known ensembles that use only the DS algorithm. The proposed ensemble can achieve an increase in correlation coefficient about 30% compared to simple DS.

C. Using ConjunctiveRule as base learner

Thirdly, we used a rule-based algorithm as base learner in the ensemble. ConjunctiveRule algorithm [13] implements a single conjunctive rule learner that can predict numeric class values. A rule consists of antecedents "AND"ed together and the consequent (class value) for the regression. In this algorithm, the consequent is the mean value in the dataset. If the test instance is not covered by this rule, then it's predicted using the default mean value of the data. This learner selects an antecedent by computing the Information of each antecendent and prunes the generated rule using Reduced Error Prunning (REP) or simple pre-pruning based on the number of antecedents. The Information is the weighted average of the mean-squared errors of both the data covered and not covered by the rule. In pruning, the weighted average of the mean-squared errors on the pruning data is used.

TABLE IV
COMPARING THE PROPOSED ENSEMBLE WITH OTHER WELL KNOWN ENSEMBLES THAT USES AS BASE LEARNER THE CONJUNCTIVERULE

Datasets	Average B&B Conjunctive-Rule	Bagging Conjunctive-Rule		Additive Regression Conjunctive-Rule		Conjunctive-Rule	
auto93	0.75	0.76		0.61	*	0.49	*
autoHorse	0.86	0.83	*	0.84	*	0.72	*
autoMpg	0.84	0.78	*	0.85	v	0.74	*
autoPrice	0.88	0.84	*	0.87	*	0.80	*
bodyfat	0.94	0.85	*	0.96	v	0.82	*
bolts	0.78	0.76		0.73	*	0.64	*
breastTumor	0.21	0.23		0.12	*	0.10	*
cholesterol	0.09	0.12	v	0.03	*	0.01	*
cleveland	0.63	0.64		0.56	*	0.42	*
cpu	0.91	0.89	*	0.86	*	0.36	*
detroit	0.18	0.19		0.07	*	0.06	*
elusage	0.84	0.85	v	0.77	*	0.69	*
fishcatch	0.91	0.86	*	0.91		0.82	*
gascons	0.83	0.75	*	0.79	*	0.66	*
housing	0.85	0.77	*	0.84		0.65	*
hungarian	0.62	0.62		0.60	*	0.55	*
longley	0.49	0.50		0.41	*	0.31	*
lowbwt	0.78	0.78		0.77	*	0.78	
pbc	0.49	0.49		0.44	*	0.39	*
pharynx	0.69	0.67	*	0.69		0.67	*
pollution	0.57	0.63	v	0.38	*	0.26	*

pwLinear	0.82	0.72	*	0.81		0.68	*
quake	0.08	0.09	v	0.05	*	0.06	*
sensory	0.31	0.29	*	0.30	*	0.28	*
servo	0.81	0.79	*	0.81		0.79	*
sleep	0.53	0.58	v	0.39	*	0.37	*
strike	0.45	0.42	*	0.34	*	0.18	*
veteran	0.35	0.35		0.18	*	0.12	*
vineyard	0.68	0.68		0.46	*	0.28	*
W-D-L		5/11/13		2/5/22		0/1/28	
Average correlation coefficient	0.63	0.61		0.57		0.47	

We compare the presented methodology with bagging, and boosting version of ConjunctiveRule (using 25 sub-learners). In the last raw of the Table IV one can see the aggregated results.

The presented ensemble is significantly more accurate than single ConjunctiveRule in 28 out of the 29 data sets, while it has significantly lower correlation coefficient in none data set. In addition, the presented ensemble is significantly more accurate than Bagging ConjunctiveRule in 13 out of the 29 data sets, whilst it has significantly lower correlation coefficient in 5 data sets. Furthermore, the proposed ensemble has significantly higher correlation coefficient in 22 out of the 29 data sets than Boost ConjunctiveRule, whereas it is significantly less accurate in 2 data sets.

To sum up, the performance of the presented ensemble is more accurate than the other well-known ensembles that use only the ConjunctiveRule algorithm. The proposed ensemble can achieve an increase in correlation coefficient about 34% compared to simple ConjunctiveRule.

In general for all tested base learners the proposed ensemble achieved higher correlation coefficient than either boosting and bagging 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 learners is a set of learners whose individual decisions are combined in some way (typically by weighted or unweighted averaging method) to predict the values of new examples. One of the most active areas of research in supervised learning has been to study methods for constructing good ensembles of learners. The main discovery is that ensembles are often much more accurate than the individual learners 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, regression 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 an averaging 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 higher correlation coefficient 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 accuracy of the order of 2% to 34% compared to the tested base learners.

Our approach answers to some extent such questions as generating uncorrelated learners and control the number of learners needed to improve accuracy in the ensemble of learners. While ensembles provide very accurate regression models, too many learners 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-learners to 10 in each sub-ensemble.

Finally, there are some open problems in ensemble of learners, such as how to understand and interpret the decision made by an ensemble of learners because an ensemble provides little insight into how it makes its decision. For learning tasks such as data mining applications where comprehensibility is crucial, averaging methods normally result in incomprehensible learner 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] Avnimelech R. and Intrator N. (1999), Boosting regression estimators. Neural Computation, 11, 491–513.
[2] Bauer, E. & Kohavi, R. (1999). An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. Machine Learning, Vol. 36, pp. 105–139.
[3] Blake, C.L. & Merz, C.J. (1998). UCI Repository of machine learning databases. Irvine, CA: University of California, Department of Information and Computer Science.
[http://www.ics.uci.edu/~mlearn/MLRepository.html]
[4] Breiman L. (1996). Bagging Predictors. Machine Learning, Vol. 24, No. 3, pp. 123-140.
[5] Breiman, L. (2001), Using Iterated Bagging to Debias Regressions, Machine Learning, Volume 45, Issue 3, Dec 2001, Pages 261 – 277.
[6] Brown, G., Wyatt, J., and P. Tino (2005). Managing diversity in regression ensembles. Journal of Machine Learning Research, 6, 2005.
[7] Dietterich, T.G. (2001). Ensemble methods in machine learning. In Kittler, J., Roli, F., eds.: Multiple Classifier Systems. LNCS Vol. 1857, pp. 1-15.
[8] Duffy N., Helmbold, D. (2002), Boosting Methods for Regression, Machine Learning, Volume 47, Issue 2 – 3, 153 – 200.
[9] Freund Y. and Robert E. Schapire (1996). Experiments with a New Boosting Algorithm, in proceedings of ICML'96, pp. 148-156.
[10] Iba, W., & Langley, P. (1992). Induction of one-level decision trees, in proceedings of Ninth International Machine Learning Conference (1992). Aberdeen, Scotland.
[11] Nadeau, C., Bengio, Y. (2003), Inference for the Generalization Error. Machine Learning, 52, 239-281.

Open Science Index, Computer and Information Engineering Vol:1, No:1, 2007 publications.waset.org/6392.pdf

[12] Opitz D. & Maclin R. (1999). Popular Ensemble Methods: An Empirical Study, Artificial Intelligence Research, Vol. 11, pp. 169-198.
[13] Witten, I. and Frank, E., Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann, San Mateo, CA, 2000.
[14] Zemel, R., Pitassi, T. (2001), A gradient-based boosting algorithm for regression problems. In T. K. Leen, T. G. Dietterich, & V. Tresp (Eds.), Advances in Neural Information Precessing Systems 13 (pp. 696–702). Cambridge, MA: MIT Press.

Sotiris Kotsiantis received a diploma in mathematics, a Master and a Ph.D. degree in computer science from the University of Patras, Greece. He is an adjunct lecturer in the Department of Computer Science and Technology at the University of Peloponnese, Greece. He is also post-doc researcher in the University of Patras, Greece. His main research interests are in the field of statistical learning, data mining and knowledge representation. He has about 65 publications to his credit in international journals and conferences.

Open Science Index, Computer and Information Engineering Vol:1, No:1, 2007 publications.waset.org/6392.pdf