Volume 3, Issue 10, October 2013 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Modified Boosting for Unbiased Classification

Nahush P Patil*, Karan Shah, Sapna Gala

Mumbai University,
India

***Abstract*— Classification of data plays a crucial role in the field of data mining. The size of the data decreases the performance and efficiency of classifier. The decreasing performance of classifier compromised with unvoted data of classifier. So the merging of two or more classifier is done for better prediction, such techniques are called Ensemble classifier. One such approach is using bagging and boosting. This paper deals with enhancing the accuracy of prediction by presenting an ensemble method which is modification to boosting, which uses the dataset as the test set instead of the training set for error calculation during derivation of the model.**

***Keywords*— Classifier, Ensemble, Boosting, Bagging.**

I. Introduction

Machine learning has many applications in data mining. Data mining involves analysis of data to outline some relationship among multiple features of data. Machine learning is applied to these process of finding relationships to improve the efficiency of system and design of machine. When instances with known label are given the learning is called supervised learning and if instances are unlabeled the learning is called unsupervised learning. But unsupervised learning provides useful classes of items which is called clusters[1]. Clusters are groups of similar types of objects. These groups are formed with classification methods[2]. These classifications are done by classifiers. Classification is a form of data analysis that extends the models describing important data classes. Such models called classifiers, predicts categorical class labels. Classification and prediction are two forms of data analysis that can be used to extract models describing important data classes or to predict future data trends. Classification is a two-step process, consisting of learning step (where a classification model is constructed) and classification step (where the model is used to predict class labels for given data). It is very important that the classification predicts the data trends accurately[3]. Ensemble methods have been called the most influential development in Data Mining and Machine Learning in the past decade. They combine multiple models into one usually more accurate than the best of its components[4]. The term "ensemble methods" is commonly reserved for bundled fits produced by a stochastic algorithm, the output of which is some combination of a large number of passes through the data. Such methods are loosely related to iterative procedures on the one hand and to bootstrap procedures on the other. An example is the average of a large number of kernel smoothes of a given variable, each based on a bootstrap sample from the same data set. The idea is that a "weak" procedure can be strengthened if given a chance to operate "by committee." Ensemble methods often perform extremely well and in many cases, can be shown to have desirable statistical properties[5]. Two of the most popular ensemble learning algorithms are Bagging and Boosting.

II. Related Work

A. Bagging

Bagging is a method for generating multiple versions of a predictor and using these to get an aggregated predictor. The aggregation averages over the versions when predicting a numerical outcome and does a plurality vote when predicting a class. The multiple versions are formed by making bootstrap replicates of the learning set(62%) and using these as new learning sets. It uses each of them to generate a classifier for inclusion in the ensemble[4]. The left (37%) are used to form accurate estimates of important quantities. For instance, they can be used to give much improved estimates of node probabilities and node error rates in decision trees. Using estimated outputs instead of the observed outputs improves accuracy in regression trees. They can also be used to give nearly optimal estimates of generalization errors for bagged predictor. Tests on real and simulated data sets using classification and regression trees and subset selection in linear regression show that bagging can give substantial gains in accuracy. The vital element is the instability of the prediction method. If perturbing the learning set can cause significant changes in the predictor constructed, then bagging can improve accuracy[8]. As shown in Fig 1 In bagging, Given a set, D, of d tuples, For iteration i (i = 1, 2, : : : , k), a training set Di, of d tuples is sampled with replacement from the original set of tuples D. Because sampling with replacement is used, some of the original tuples of D may not be included in Di, whereas others may occur more than once. A classifier model, Mi, is learned for each training set, Di. To classify an unknown tuple, **X**, each classifier, Mi, returns its class prediction, which counts as one vote. The bagged classifier, M*, counts the votes and assigns the class with the most votes to **X**.

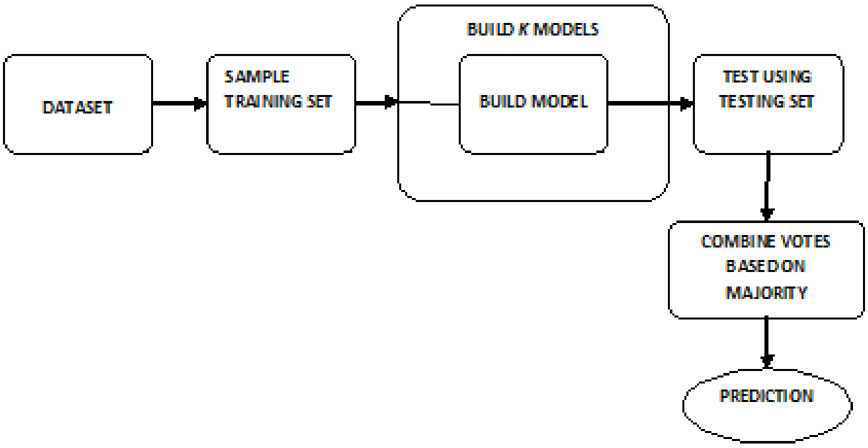

Fig 1. Bagging Model.

Bagging can be applied to the prediction of continuous values by taking the average value of each prediction for a given test tuple. The bagged classifier often has significantly greater accuracy than a single classifier derived from D, the original training data. It will not be considerably worse and is more robust to the effects of noisy data. The increased accuracy occurs because the composite model reduces the variance of the individual classifiers. For prediction, it was theoretically proven that a bagged predictor will always have improved accuracy over a single predictor derived from D[3]. In bagging, the M training sets that are created are likely to have some differences. If these differences are enough to induce noticeable differences among the M base models while leaving their performances reasonably good, then the ensemble will probably perform better than the base models individually[6].

B. Boosting

Boosting refers to a general and provably effective method of producing a very accurate prediction rule. One of the main ideas of the algorithm is to maintain a distribution or set of weights over the training set. Initially, The weight of this distribution on training set and all weights are set equally, but on each round, the weights of incorrectly classified examples are increased so that the weak learner is forced to focus on the hard examples in the training set[7]. The higher the weight, the more the instance influences the classifier learned. It is a powerful technique that can usually produce better ensembles than Bagging[9]. Fig 2 shows a working model of boosting ensemble.

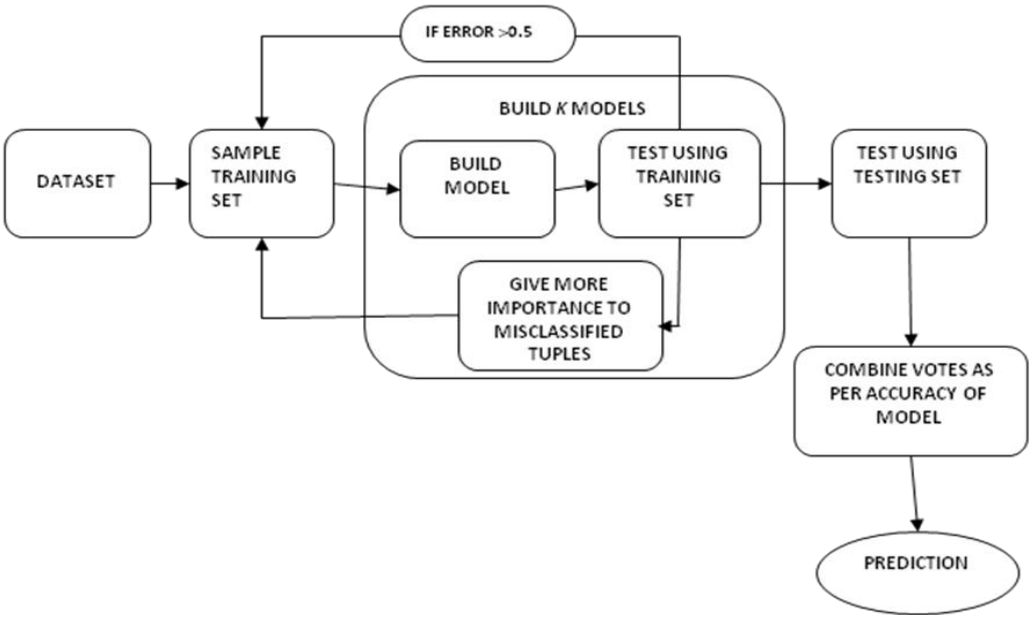

Fig 2. Boosting Model.

In boosting, weights are assigned to each training tuple. A series of k classifiers is iteratively learned. After a classifier Mi is learned, the weights are updated to allow the subsequent classifier,Mi+1, to “pay more attention” to the training tuples that were misclassified by Mi. The final boosted classifier, M*, combines the votes of each individual classifier, where the weight of each classifier's vote is a function of its accuracy. We are given D, a data set of d class-labeled tuples, (**X1**, y1), (**X2**, y2), .., (**Xd**, yd),where yi is the class label of tuple **Xi**. Initially, AdaBoost assigns each training tuple an equal weight of 1=d. Generating k classifiers for the ensemble requires k rounds through the rest of the algorithm. In round i, the tuples from D are sampled to form a training set Di, of size d. Sampling with replacement is used—the same tuple may be selected more than once. Each tuple's chance of being selected is based on its weight. A classifier model, Mi, is derived from the training tuples of Di. Its error is then calculated using Di as a test set. The weights of the training tuples are then adjusted according to how they were classified. If a tuple was incorrectly classified, its weight is increased. If a tuple was correctly classified, its weight is decreased. A tuple's weight reflects how hard it is to classify—the higher the weight, the more often it has been misclassified. These weights will be used to generate the training samples for the classifier of the next round[3]. The actual performance of boosting on a particular problem is clearly dependent on the data and the weak learner. Consistent with theory, boosting can fail to perform well given insufficient data, overly complex weak hypotheses or weak hypotheses which are too weak. Boosting seems to be especially susceptible to noise. The AdaBoost algorithm, solved many of the practical difficulties of the earlier boosting algorithms. AdaBoost has many advantages. It is fast, simple and easy to program. It has no parameters to tune (except for the number of round). It requires no prior knowledge about the weak learner and so can be flexibly combined with any method for finding weak hypotheses. Finally, it comes with a set of theoretical guarantees given sufficient data and a weak learner that can reliably provide only moderately accurate weak hypotheses. This is a shift in mind set for the learning-system designer: instead of trying to design a learning algorithm that is accurate over the entire space[7].

III. PROPOSED SYSTEM

Modified boosting (ModBoost) provides an improved version of classifier as shown in fig 3. Given *D*, a data set of *d* class-labeled tuples. In each round *i*, the tuples from *D* are sampled to form a training set *Di*, of size *d*. Sampling with replacement is used—the same tuple may be selected more than once. Each tuple's chance of being selected is based on its weight. A classifier model, *Mi*, is derived from the training tuples of *Di*. Its error is then calculated using dataset D, as a test set , unlike boosting. The weights of the training tuples are then adjusted according to how they were classified. If a tuple was incorrectly classified, its weight is increased. If a tuple was correctly classified, its weight is decreased. A tuple's weight reflects how hard it is to classify—the higher the weight, the more often it has been misclassified. These weights will be used to generate the training samples for the classifier of the next round. The correctly classified Di tuples are then tested again using the testing set. The lower a classifier's error rate, the more accurate it is. It is assigned as a weight to each classifier's vote, based on how well the classifier performed. The votes are then combined which acts as a function of its accuracy for prediction.

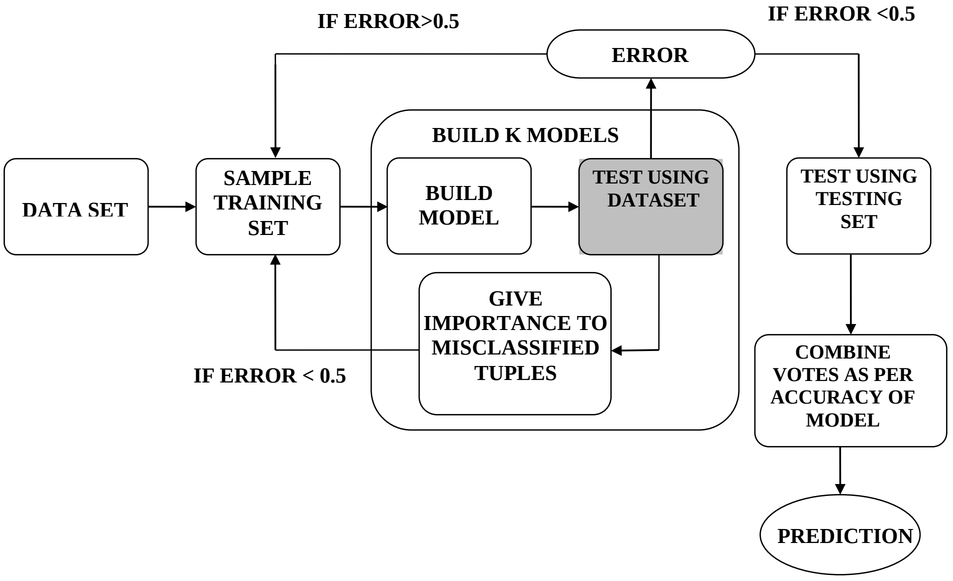

Fig 3. Modified Boosting Model.

ModBoost Algorithm:

1) initialize the weight of each tuple in *D* to 1=*d*;
2) for *i* = 1 to *k* do // for each round:
3) sample *D* with replacement according to the tuple weights to obtain *Di*;
4) use training set *Di* to derive a model, *Mi*;
5) compute *error*(*Mi*) using dataset D, the error rate of *Mi*
6) if *error*(*Mi*) > 0:5 then
7) reinitialize the weights to 1=*d*
8) go back to step 3 and try again

```
9)  endif
10) for each tuple in $Di$ that was correctly classified do
11) multiply the weight of the tuple by $error(Mi)$=(1-$error(Mi)$); // update weights
12) normalize the weight of each tuple
13) compute accuracy using testing set;
14) endfor
```

IV. ANALYSIS

As discussed, In bagging, the test set which is independent of training tuples is used as testing set to test the classifier for accuracy. The classifier is trained on a particular set of data and tested on a different one. Whereas, In boosting the training set itself is used as testing set to test the classifier. This makes the approach pessimistic towards calculating the efficiency. In modified boosting (ModBoost) we test the model with testing set and data set. Testing the model with both, the known values and unknown values , makes this approach more realistic and unbiased.

V. CONCLUSION AND FUTURE WORK

Ensemble Methods are motivated by the idea of wanting to leverage the power of multiple models with Significant theoretical and experimental developments and not just trust one model built on a small training set. Our study was meant to provide one such contribution. In this paper we review various methods of ensemble classifier and propose an enhancement technique. The method is complex, but this approach is justified by the better results that can be assessed. Our further work is to find out its merits and applications in other areas. It is being expected to achieve a robust ensemble with an improved accuracy for better prediction.

ACKNOWLEDGEMENT

The authors would like to thank Mrs. Swati Deshpande, IT Head of Department-Shah And Anchor Kutchhi Engineering college, for her constant encouragement and support towards achieving this goal. The authors express deepest gratitude to Ms. Kranti Ghag for invaluable suggestions. Without her guidance and persistent help this dissertation would not have been possible.

REFERENCES

[1] Xueyi Wang "*A New Model for Measuring the Accuracies*" in IEEE World Congress on Computational Intelligence, 2012.

[2] Tao Dacheng, Tang Xiaoou, Li Xuelong Wu and Xindong . '*Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval*" in IEEE Transactions, 2006.

[3] J. Han and M. Kamber, '*Data Mining: Concepts and Techniques'*, 2nd ed. pp 285-367,2000.

[4] Nikunj C. Oza, Ph.D , *Ensemble Data Mining Methods*, NASA Ames Research Center, USA, 2005.

[5] Breiman, Leo. '*Random Forest'*, Statistics Department. University of California. Berkeley, CA. 94720,2001.

[6] Breiman Leo, '*Bagging predictors'*, Technical Report Statistics Department, UCB,1994.

[7] Yoav Freund and Robert E. Schapire,*'A Short Issntroduction to Boosting'*, AT&T Labs, Research, Shannon Laboratory, USA,1999.

[8] Breiman Leo, '*Bagging Predictors [1996a]' Machine Learning* 26: pp 123–40,1996.

[9] David Opitz and Richard Maclin, *An Empirical Evaluation of Bagging and Boosting*, University of Minnesota-Duluth,1998.