

An Experiment to Improve Classification Accuracy Using Ensemble Methods

Bhavesh Patankar*¹, Dr. Vijay Chavda²

*¹Department of M.Sc. (IT), Kadi SarvaVishwaVidyalaya, Gandhinagar, Gujarat, India.

²NPCCSM, Kadi SarvaVishwaVidyalaya, Gandhinagar, Gujarat, India

ABSTRACT

Data mining is the practice of analyzing huge quantities of data and shortening it into constructive knowledge. Data Mining is an eternal process which is quite useful in finding understandable patterns and relationships amongst the data. There are various classification techniques available. It is observed that all the techniques don't work well with all datasets. It is found that when the classifiers are used alone, they are not performing as good as when they are combined using ensembles. Ensemble methods are renowned techniques in order to improve the classification accuracy. Bagging and Boosting are the most common ensemble learning techniques used to improve the classification accuracy. Here, a study on the classification accuracy improvement is carried out in which an experiment is performed using boosting with different datasets from UCI repository.

Keywords: Data mining; classification; ensemble learning; boosting, Adaboost;

I. INTRODUCTION

Data mining refers to extracting knowledge from large amounts of data available from various data sources which are accumulated in data warehouse. It is an interdisciplinary field, which covers variety of areas like data warehousing, statistical methods, database management systems, artificial intelligence, information retrieval, data visualization, pattern recognition, spatial data study, digital signal processing, image databases and many more other application fields, like business, economics, and bioinformatics.[1]

In data mining, Classification is a classical problem extensively studied by statisticians and machine learning Researchers. Classification is tasks in which we have to find patterns. In Classification, first of all model construction is carried out after that model usage is done. The general steps in classification are given below.

1. Create a model from training data. This usually involves an algorithm that searches for the "best fit" to the training data.

2. Evaluate the model on test data. This usually involves estimating the accuracy of the model.
3. Apply an acceptable model on the target datasets.

Accuracy of classification is one of the important features. To improve the classification accuracy, various strategies have been identified. Ensemble learning is one of the ways to improve the classification accuracy. Ensembles are learning techniques that builds a set of classifiers and then classify new data sets on the basis of their weighted vote of predictions. The ensemble learning techniques include Bagging, boosting etc.[2]. In this paper reviews for these methods have been made and explained why ensembles can often perform better than single classifiers. Combining outputs from multiple classifiers, known as ensemble learning, is one of the standard and most important techniques for improving classification accuracy in machine learning. Out of these, bagging and boosting are the most popular methods of ensemble learning. In bagging, from the training data, a training set is randomly sampled k times with replacement which produces k training sets with exactly

the same size as what we have in original training set. As the original data set is sampled with replacement, it may happen that some training instances are repeated in the new training sets, and it is quite possible that some are not present at all. The obtained sample sets are used to train base classifiers like CART etc. which in turn will give k different predictors. These k different predictors are used to classify the new dataset.

The classification for each data instance is obtained by equal weight voting on all k predictors. Voting gives a significant improvement in classification accuracy and stability. Boosting, on the other hand, induces the ensemble of classifiers by adaptively changing the distribution of the training set based on the accuracy of the previously created classifiers and uses a measure of classifier performance to weight the selection of training examples and the voting.

Various empirical studies, suggest that combining classifiers gives optimal improvements in accuracy if the classifiers are not correlated. It is stated in Ref. [3], the most effective method of achieving such autonomy is by training the members of an ensemble on qualitatively different feature (sub) sets. In other words, attribute partitioning methods are capable of performance superior to data partitioning methods (e.g. bagging and boosting) in ensemble learning. There are a growing number of publications that investigate performance of classifier ensembles trained using attribute

AdaBoost is a one of the ensemble learning which is more practical towards the boosting approach. Adaboost requires lesser instability than bagging, because Adaboost can make much larger changes in the training set.[5] A number of studies that compare AdaBoost and bagging suggest that AdaBoost and bagging have quite different operational profiles (Bauer and Kohavi 1999; Quinlan 1996). In general, it is found that bagging is more consistent, increasing the error of the base learner less frequently than does AdaBoost. However, AdaBoost appears to have greater average effect, leading to substantially larger error reductions than bagging on average.

Generally, bagging tends to decrease variance without unduly affecting bias (Breiman 1996; Schapire et al. 1998; Bauer and Kohavi 1999). On the contrary, in empirical studies AdaBoost appears to reduce both bias and variance (Breiman 1996; Schapire et al. 1998; Bauer

and Kohavi 1999). Thus, AdaBoost is more effective at reducing bias than bagging, but bagging is more effective than AdaBoost at reducing variance.

The decision on limiting the number of sub-classifiers is important for practical applications. To be competitive, it is important that the algorithms run in reasonable time.

Tu. et. al[3] proposed the use of bagging with decision tree C4.5 algorithm and with Naive Bayes algorithm to identify the heart disease of a patient and compare the effectiveness, correction rate among them. They studied the data collected from patients with coronary artery disease.

Kittler et. al[4] has developed a common theoretical framework for combining classifiers which use different pattern representations and show that many existing schemes can be considered as special cases of compound classification where all the pattern representations are used jointly to make a decision. Comparison of various classifier combination schemes demonstrates that the combination rule developed under the most restrictive assumptions-the sum rule-outperforms other classifier combinations schemes. An analysis of the various schemes to estimation errors is carried out to show that this finding can be justified theoretically.

Nguyen et. al[5] have compared two classifiers (decision tree and Bayesian network) to predict students GPA at the end of the third year of undergraduate and at the end of the first year of postgraduate from two different institutes. Each data set has 20,492 and 936 complete student records respectively. The results show that the decision tree outperformed Bayesian network in all classes. The accuracy was further improved by using re-sampling technique especially for decision tree in all cases of classes. In the same time it able to reduce misclassification especially on minority class of imbalanced datasets because decision tree algorithm tends to focus on local optimum.

Thomas [6] performed experiments which show that in situations where there is little or no classification noise, randomization is competitive with (and perhaps slightly superior to) bagging but not as accurate as boosting. In situations with considerable classification noise, it is found that bagging is much better than boosting.

II. METHODS AND MATERIAL

Boosting

Boosting is a practical approach towards machine learning based on the idea of creating a highly accurate prediction regulation by combining many relatively weak and inaccurate regulations.

The AdaBoost algorithm of Freund and Schapire [7] was the first practical boosting algorithm, and still remains one of the most widely used and studied, with applications in various fields. Over the years, a great variety of attempts have been made to “explain” AdaBoost as a learning algorithm, that is, to understand why it works, how it works, and when it works (or fails). AdaBoost generally used to boost weak learning algorithm into strong learning algorithm. AdaBoost generates an ensemble of classifiers, the training data of each is drawn from a distribution that starts uniform and iteratively changes into one that provides more weight to those instances that are misclassified. Each classifier in AdaBoost focuses increasingly on the more difficult to classify instances. The classifiers are then combined through weighted majority voting.

AdaBoost is a popular boosting algorithm. Let say, we need to boost the accuracy of some learning routine. Also, it is given D , a dataset of d class labeled tuples $(A_1, b_1), (A_2, b_2), (A_3, b_3), \dots, (A_d, b_d)$, where y_i is the class label of tuple A_i . Firstly, AdaBoost assign each training tuple an equal weight of $1/d$. In order to generate k classifiers for the ensemble requires k rounds throughout the rest of the algorithm. In the round i , the tuples from D are sampled to make a training set, D_i , of size d . sampling with replacement is used – the same tuple may be selected more than once. The probability of each tuple of being chosen is based on its weight. A classifier, M_i , is consequent from the training tuples of D_i . Its error is then calculated using D_i as a test set. The weights of the training tuples are then accustomed according to how they were classified. If a tuple were wrongly classified, its weight is augmented. If a tuple was appropriately classified, its weight is reduced. A tuple’s weight reflects how hard it is to classify – the higher the weight, chances of misclassification is higher. These weights will be used to produce the training samples for the classifier of the next iteration. The general thought is that when we build a classifier, we

want it to focus more on the misclassified tuples of the previous iteration. Some classifiers may be superior at classifying some “hard” tuples than others. In this way, we build a series of classifiers that harmonize each other [8].

III. RESULT AND DISCUSSION

We have chosen Weka tool to perform the experiment. Weka (Waikato Environment for Knowledge Analysis) is a popular machine learning tool developed in JAVA. It is one of the free open source softwares available under the GNU General Public License. Here, the experiment is performed on base classifier and then accuracy is measured. After that experiment is performed on the classifier with boosting. The experiment is carried out using dataset collected from UCI machine repository. Lastly, results are compared and conclusion is derived.

In our experiment, we’ve taken following datasets from the UCI Machine Learning Repository.

Sr.No	Dataset Information		
	Dataset	Instances	Attributes
1	Iris	150	5
2	Diabetes	768	9
3	Breast-Cancer	286	10

The experiment is carried out on RepTree, Decision Sump and J48 classifier. The datasets are chosen and no filter is applied while carrying out the experiment. Firstly experiment is carried out using single base classifier then experiment is carried out using single base classifier with bagging. The experiment is carried out using weak 3.6.12.

Accuracy of the base single classifier and base classifier with bagging is measured which is displayed in below table.

Classifier	Datasets		
	Iris	Diabetes	Breast-Cancer
Decisionsump	66.67	71.87	68.53

Classifier	Datasets		
	<i>Iris</i>	<i>Diabetes</i>	<i>Breast-Cancer</i>
Decisionsump with AdaBoost	95.33	74.34	70.27
OneR	92.00	71.48	65.73
OneR with AdaBoost	92.67	70.18	68.53

We can see the result of the classifiers when used alone and when used with boostig. The columnar chart clearly shows the effect of base classifier with bagging.

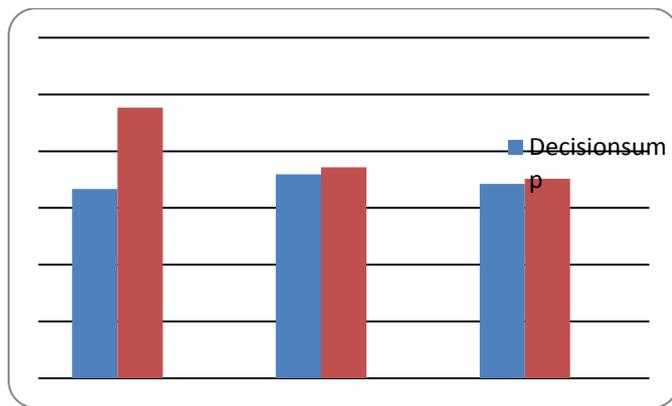


Figure 1: Decisionsump and Decisionsump with AdaBoost comparison

It is clearly seen that when Decisionsump is used alone with iris, diabetes and breast-cancer dataset, the accuracy of classifier is lesser than when it is used with AdaBoost.

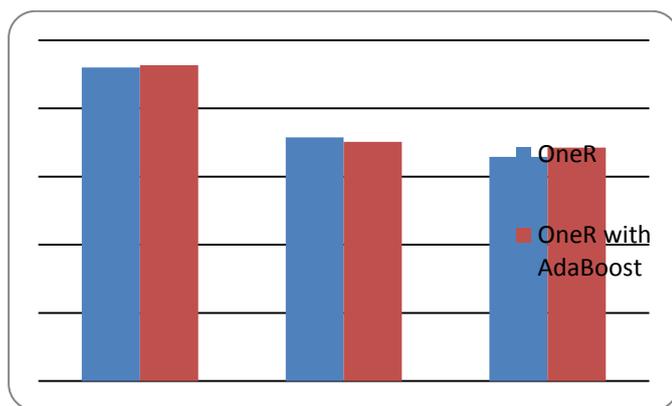


Figure 2: OneR comparison with OneR with AdaBoost

It is clearly seen that OneR is used alone with iris and breast-cancer dataset, the accuracy of classifier is lesser than when it is used with AdaBoost. Here one exception is there that is when same thing is performed with

Diabetes dataset the ensemble accuracy goes down. So from above experiment, we can say that boosting improves the classification accuracy.

IV. CONCLUSION

The paper shows the effect of boosting (here AdaBoostM1) on classification accuracy by using different classifiers. The experiment was carried out using weak 3.6.12 and showed the effect of AdaBoostM1 on various base classifiers. Adding to it, it was observed that for all the three datasets, the classification accuracy increases when we use ensemble learning instead of a single classifier, exception was the diabetes dataset with OneR classifier. In conclusion, ensemble learning technique of boosting helps in improving the accuracy of classification. Future directions can include the effects of changing the base classifier learner like naive bayes, neural network, CART etc.

V. REFERENCES

- [1] Han, Jiawei, and Micheline Kamber. "Data mining: concepts and techniques (the Morgan Kaufmann Series in data management systems)." (2000).
- [2] Dietterich, Thomas G. "Ensemble methods in machine learning." Multiple classifier systems. Springer Berlin Heidelberg, 2000. 1-15.
- [3] Tu, My Chau, Dongil Shin, and Dongkyoo Shin. "A comparative study of medical data classification methods based on decision tree and bagging algorithms." Dependable, Autonomic and Secure Computing, 2009. DASC'09. Eighth IEEE International Conference on. IEEE, 2009.
- [4] Kittler, Josef, et al. "On combining classifiers." Pattern Analysis and Machine Intelligence, IEEE Transactions on 20.3 (1998): 226-239.
- [5] Nguyen Thai Nghe, P. Janecek, and P. Haddawy, "A comparative analysis of techniques for predicting academic performance", ASEE/IEEE Frontiers in Education Conference, pp. T2G7-T2G12, 2007.
- [6] Dietterich, Thomas G. "An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization." Machine learning 40.2 (2000): 139-157.
- [7] Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences 55(1), 119–139 (1997)
- [8] Oza, N.C. AveBoost2: Boosting with Noisy Data. In F. Roli , J. Kittler, and T. Windeatt (Eds.), Proceedings of the Fifth International Workshop on Multiple Classifier Systems, 31-40, Berlin, 2004.