Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 5, May 2014, pg.727 – 732

RESEARCH ARTICLE

Improving Classification Accuracy Using Ensemble Learning Technique (Using Different Decision Trees)

Nikita Joshi[1], Shweta Srivastava[2]

[1]Assistant Professor, Department of Computer Science & Engineering, ABES Engineering College, Ghaziabad, India
[2]Assistant Professor, Department of Computer Science & Engineering, ABES Engineering College, Ghaziabad, India
[1] nikita.joshi@abes.ac.in ; [2] shweta.shrivastava@abes.ac.in

Abstract- Using ensemble methods is one of the general strategies to improve the accuracy of classifier and predictor. Bagging is one of the suitable ensemble learning methods. Ensemble learning is a simple, useful and effective meta-classification methodology that combines the predictions from multiple base classifiers (or learners). In this paper we show a comparative study of different classifiers (Decision trees) when the ensemble learning technique called bagging is used. We perform classification on various datasets firstly by using a single classifier and then by bagging method, using the same base classifier. It is observed that when we use a single classifier rather than an ensemble, the classification error further increases. As different training data subsets are randomly drawn-with-replacement from the entire training dataset, so usually the new training set contains some duplicates and some omissions as compared to the original training set. Each training data subset is used to train a different classifier of the same type.

Keywords – Ensemble Learning; decision tree; bagging; decision tree inducer; classification accuracy

I. INTRODUCTION

Ensemble Learning is a two-step decision making process, in which the first step is related to the decision of the individual classifier and the second step refers to the decision of the combined model. In recent years, various methods for creation of an ensemble have been proposed, which have proved out to be very useful for machine learning and pattern recognition [1], [2], [9], [4]. Ensemble methods create base classifiers and the outputs are combined, usually by voting, to get better classification accuracy. Improved classification results can be achieved by using diverse classifiers. Bagging [4], Boosting [6] and Rotation Forest [7] are three of the most famous ensemble creation methods [3]. In Bagging, a number of patterns are randomly selected from the original training set with replacement. New training set has the same number of patterns as the original one with some repetitions and some omissions. The new training set is called a “Bootstrap Replicate” of the original training set. After training, the ensemble classifies testing set by using Majority vote. A.Rokach[8] has proposed a number of majority voting algorithms. Various methods for creating ensemble classifiers have been reviewed and proved out to be helpful for improving the accuracy of classifiers. J. Franke [6] explores the bootstrapping classification method in the context of artificial neural networks.

E. Bauer [7] reviews several voting classification algorithms, and compares several variants in association with a decision tree inducer and a Naive-Bayes inducer. Fumera [5] applied an analytical framework for the analysis of linearly combined classifiers to ensembles generated by bagging.

II. BACKGROUND

A. Bagging

Bagging, introduced by Breiman [4], takes bootstrap samples of objects and classifiers are trained on each sample. The classifier votes are then combined by majority voting. Breiman shows that the predictors build using Bagging can make an unstable weak classifier significantly optimal. The Bagging predictors have been used in various classification and prediction problems in biology and social sciences.

Diverse classifiers are generated by randomly selecting subsets of samples to train them .Bagging procedure is shown in Fig.1. Given a training dataset of size N, Bagging creates M base models, each trained on a bootstrap sample of size N created by drawing random samples with replacement from the original training set.

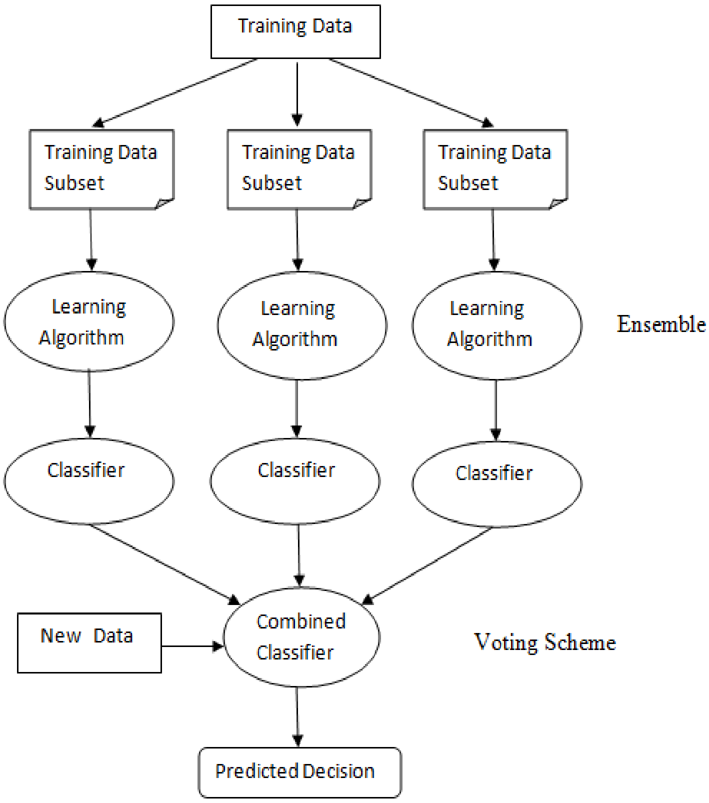

Fig.1. Bagging procedure

The pseudocode for the batch bagging is given in Fig. 2 below.

Bagging (T, M)

- For each m ε {1,2,…….M},
  T_m = Sample_with_Replacement (T, N)
  $h_m = L_b (T_m)$
- Return { $h_1, h_2, h_3 \ldots\ldots\ldots h_M$ }

Fig. 2 Pseudocode for Batch Bagging [10]

In the pseudocode for Batch Bagging in Figure 2., M is the number of base models to be learned, T is the original training set of N examples , L_b is the base model learning algorithm, the $h_{i's}$ are the base learner model. A function h(x) is returned that classifies new examples by returning the class Y that gets the maximum number of votes from the base models h_1, h_2..... h_M

B. Decision Tree

Decision tree learning is one of the most popular technique in classification as it is fast and produces models having fair performance. Based on given input variables, a decision tree creates a model that predicts the value of a target variable. Each internal node represents a test on an attribute value. Each branch corresponds to an attribute value. A leaf node contains the classification to be returned if it is reached. Decision trees give impressive predictions and provide an explicit concept description for a dataset. Decision trees are constructed in a top-down manner.

C. Decision Tree Inducers

Several algorithms exists which automatically construct a decision tree from a given dataset. Such algorithms are known as Decision tree inducers. Their goal is to find the optimal decision tree by considerably reducing the generalization error. These algorithms also work on reducing the number of nodes in the tree or minimizing the average depth of the decision tree. There are various decision tree inducers like ID3, C4.5, CART, decision stump and BF Tree.

1) Decision stump: It is a decision tree having one root node (internal node), connected immediately to the terminal leaf nodes. Predictions are made based on the value of a single input feature. Several variations in the predictions are possible depending on the type of the input feature. Generally for nominal features, a stump is build containing a leaf for each possible feature value. A stump with two leaves has one leaf corresponding to some selected category and the other leaf to all the other remaining categories. In some cases where there exists some continuous feature, then some threshold feature value is selected, and the two leaves of the stump correspond to values below and above the threshold. A missing value is treated as a yet another category. If a stump exists with two leaves, one of which corresponds to some chosen category, and the other leaf to all the other categories. For binary features these two schemes are identical. A missing value may be treated as a yet another category. Decision stumps are often used as weak learners [17] or base learners in machine learning ensemble techniques such as bagging and boosting.

2) CART: CART (Classification and Regression Trees) [12] constructs binary trees where each internal node has exactly two outgoing nodes whereas decision trees are generated using ID3 and C4.5 algorithms have variable branches per node. CART is used for regression analysis with the help of regression trees. In CART, trees are grown using Gini index for tree splitting. The procedure is carried out to the maximum size without using any stopping rule and the tree is then pruned back to the root node via cost-complexity pruning. The CART mechanism is intended to produce not one, but a CART generates a chain of nested pruned trees (optimal trees). The CART mechanism performs automatic class balancing, handling missing values, cost-sensitive learning, dynamic feature construction, and probability tree estimation [11]. The tree is pruned by cost–complexity Pruning. An important characteristic of CART is the generation of regression trees. Leaves of Regression trees predict a real number instead of a class. When generating regression trees, CART finds those splits which minimize the prediction squared error. The prediction made in each leaf is based on the weighted mean for the particular node.

3) C4.5: The ID3 algorithm is a very simple decision tree algorithm [13] and uses information gain as the tree splitting criteria. The growing of tree nodes ceases when all instances belong to a single value of target feature or when best information gain is less than zero. ID3 does not perform any kind of pruning procedures and neither handles numeric attributes or missing values. C4.5 is an evolution of ID3 proposed by the Quinlan [14]. It uses information gain ratio as the tree splitting criteria. The splitting stops when the number of instances to be split is below a certain threshold value. C4.5 performs error–based pruning and also handles numeric attributes. J48 is an open source Java implementation of the C4.5 algorithm in the Weka data mining tool.

4) BF Tree: Best-first decision trees are constructed in a divide-and-conquer manner. The Best-first decision trees performs the best split in the tree based on boosting algorithms[15] which is used to expand nodes in best-first order preferred to fixed order. BF Tree uses either information gain or gini index for calculating the best node in the tree in each step. The best node is the node that greatly lowers impurity among all non-terminal nodes that are available for splitting. Pre-pruning as well as post-pruning can be performed in this manner.

III. EXPERIMENT & DISCUSSION

This experiment demonstrates the process of Bagging on different datasets using various classifiers particularly decision trees. The goal of this experiment is to perform Bagging using Weka package [18] and observe the classification results using different base classifiers .The experiment was carried out using J48 (Weka implementation of C4.5) , BFTree, CART and Decision Stump as the base models and compared the results firstly by using single classifier and then by bagging method.

A. Data

The six pure continuous data sets are taken from the UCI Machine Learning Repository [16]. The information about datasets is presented in Table I.

TABLE I

SIZE OF UCI DATA SETS USED IN OUR EXPERIMENT

S.No.	Datasets	Instances	Attributes
1	Anneal	898	39
2	Credit	490	16
3	Iris	150	5
4	Wine	178	14
5	Zoo	101	18
6	Vowel	990	14
7	Dermatology	366	35

B. Experiment

The datasets are chosen and no filter is applied on them. Firstly, classification is performed using single base classifier. The Classifier model is built by the full training set. Test mode is 10-fold cross-validation. For each of the seven datasets, four different classifiers such as J48 (Weka implementation of C4.5) , BF Tree, CART and Decision Stump are chosen .The percentage of incorrectly classified instances is noted for each dataset with different classifiers. Next, the classifier chosen is bagging. The base classifier to be used in Bagging (for each dataset) is BF Tree, CART, J48 and Decision Stump. Size of each bag, as a percentage of the training set size is 100 %. The number of iterations to be performed is 10 and the percentage of correctly classified instances is noted for each dataset with different base classifiers.

C. Result

As can be seen from Table II and Fig. 3,4,5,6, for all the datasets, the classification accuracy was higher when bagging was used, instead of single classifier. Furthermore it is seen that if the bag size percent is reduced from 100%, the accuracy in classification further decreases.

Datasets / Decision Tree	Anneal	Credit	Iris	Wine	Zoo	Dermatology	Vowel
BF Tree	98.55	96.32	94	89.88	40.59	93.98	79.59
BF Tree with Bagging	99.66	95.10	94.66	94.94	98.01	95.35	86.56
J48	98.44	85.3	96	93.82	92.07	93.98	81.51
J48 with Bagging	98.88	88.16	97.33	94.32	93.06	95.35	88.48
Decision stump	77.17	86.53	66.66	57.86	60.39	50.27	17.37

Decision stump with Bagging	83.63	86.53	72	84.83	61.38	50.29	24.64
CART	98.32	85.71	95.33	89.32	40.59	93.89	80.40
CART with Bagging	98.44	86.93	95.33	95.50	98.01	95.35	87.27

Table II gives the Classification accuracy when bagging is performed using different decision trees on various datasets.

TABLE II

CLASSIFICATION ACCURACY FOR BAGGING

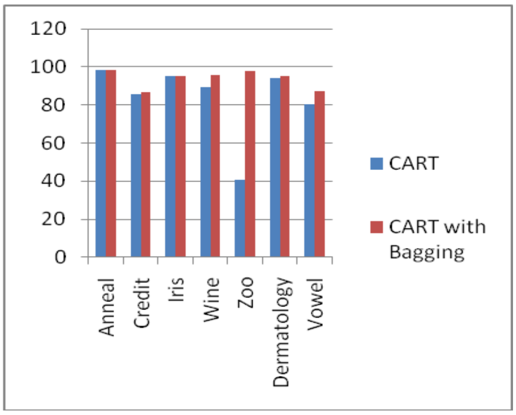

Fig.3 Comparison of Bagging vs single classifier using CART as base classifier

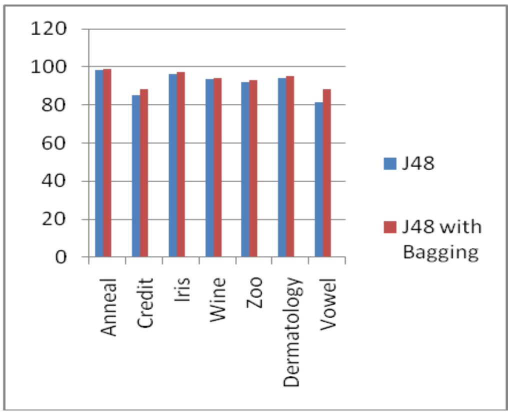

Fig.4 Comparison of Bagging vs single classifier using J48 as base classifier

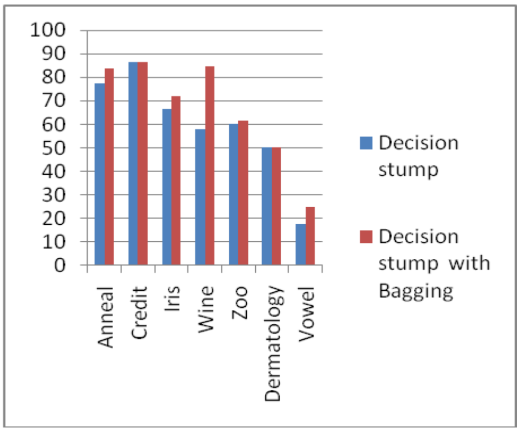

Fig.5 Comparison of Bagging vs single classifier using Decision Stump as base classifier

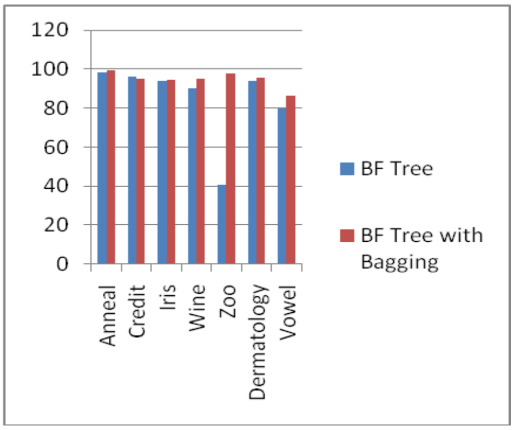

Fig.6 Comparison of Bagging vs single classifier using BF Tree as base classifier

IV. CONCLUSION AND FUTURE WORK

The paper is aimed to study and compare the effect of bagging on classification accuracy by using different decision trees as the base classifiers. The experiment shows the effect of bagging on various base classifiers. It was noticed that for all seven different datasets, the classification accuracy increases when we use ensemble learning instead of a single classifier. Even when different decision trees were used as the base classifiers, the classification accuracy was higher in case of bagging. It is concluded that ensemble learning technique of bagging helps in improving the classification accuracy. Future work can include the effects of changing the base classifier learner from decision tree to some other complex classifiers like neural network and naive bayes etc. Further study can be made on diversity and the effect of varying ensemble size on classification accuracy.

REFERENCES

[1] Nima Hatami, "Thinned-ECOC ensemble based on sequential code shrinking ", Expert Systems with Applications, vol. 39, no. 1, pp. 936-947, January 2012.
[2] Nan-Chen Hsieh and Lun-Ping Hung, "A data driven ensemble classifier for credit scoring analysis", Expert Systems with Applications, vol. 37, no. 1, pp. 534-545, January 2010.
[3]Jalil Ghavidel, Sasan Yazdani, Morteza Analoui, "A New Ensemble Classifier Creation Method by Creating New Training Set for Each Base Classifier", In Proc. 5th Conference on Information and Knowledge Technology, Iran,2013.
[4] L. Breiman, "Bagging predictors", Machine Learning, vol. 24, no. 2, pp.123-140, 1996.
[5]. Fumera, G., Roli, F., & Serrau, A., "A theoretical analysis of bagging as a linear combination of classifiers", IEEE Transactions on Pattern Analysis Machine Intelligence, Vol.30 Issue 7 , pp.1293–1299, 2008.
[6] J. Franke and M. H. Neumann, "Bootstrapping Neural Networks, Neural Computation", Volume 12 Issue 8, pp. 1929-1949, 2000.
[7] E. Bauer and R. Kohavi, "An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting,and Variants", Machine Learning,Vol.36, pp. 105 – 139, 1999.
[8] Lior Rokach, "Pattern classification using ensemble methods: World Scientific", 2010.
[9] P. Melville and R. J. Mooney, "Constructing diverse classifier ensembles using artificial training examples", International Joint Conference on Artificial Intelligence, pp. 505-512, 2003.
10] Nikunj C. Oza and Stuart Russsell, "Online Bagging and Boosting", Computer Science Division, University of California
[11].Xin dong Wu,Vipin Kumar ,J.RossQuinlan, Joydeep Ghosh,Qiang Yang, HiroshiMotoda , "Top 10 algorithms in data mining", Spinger- Verlag London Limited, pp. 20 -28, 2007.
[12] Breiman, Leo, Friedman, J. H. , Olshen,R. A. & Stone, C. J, 1984, "Classification and regression trees, Monterey", CA: Wadsworth &Brooks/Cole Advanced Books & Software, ISBN978-0412048418
[13] Quinlan, J.R., "Induction of decision trees", Machine Learning 1, pp. 81-106, 1986.
[14] Quinlan, J. R., "C4.5: Programs for Machine Learning", Morgan Kaufmann, Los Altos, 1993.
[15] J. Friedman, T. Hastie, and R. Tibshirani, "Additive Logistic Regression: a Statistical View of Boosting" , Annals of Statistics, Vol.28, pp.337-374, 2000.
[16] http://www.ics.uci.edu/~mlearn/MLRepository.html
[17] Reyzin, Lev, Schapire, Robert E ," How Boosting the Margin Can Also Boost Classifier Complexity", ICML′06: Proceedings of the 23rd international conference on Machine Learning,, pp. 753-760, 2006
[18] http://www.cs.waikato.ac.nz/ml/weka

About the Author

Ms Nikita Joshi holds an M.Tech (Computer Science and Engg.) from Amity University Uttar Pradesh and B.Tech from Uttar Pradesh Technical University. She has been working as Assistant Professor in ABES Engineering College, Ghaziabad, Uttar Pradesh, India. Her research interests include Data mining and Ensemble learning , machine translation, ,Semantic Web and wireless networks, She has extensive experience on Linux, Windows, Oracle, DB 2, Java and C++.

