International Journal of Research in Advent Technology, Vol.6, No.8, August 2018
E-ISSN: 2321-9637
Available online at www.ijrat.org

Feature Selection: Improved Random Forest Classifier

Chetna Sharma [1], Aman Kumar Sharma [2],
Department[1,2], *Department of Computer Science, Himachal Pradesh University, Shimla, India*[1,2]
Email: sharmachetna688@gmail.com [1], *sharmaas1@gmail.com* [2]

Abstract- Classification performance of random forest improves with increase in the size of forest. But experimental evidences suggest that adding trees beyond certain pre-determined limit may not significantly improve the classification performance of random forest**.** The proposed method uses feature selection methods. Feature selection methods yields a way of reducing calculation time, improving classification accuracy and a better understanding of the data in machine learning. Based on the number of important and unimportant features, a theoretical upper limit on the number of trees to be added to the forest to ensure improvement in classification accuracy. Improved Random Forest classifier is proposed which performs classification with minimum number of trees. This algorithm meets with a reduced but important set of features. It is to be proved that further addition of trees or reduction of features improve classification accuracy of the random forest.

Index Terms- Random Forest; feature selection; feature reduction; bagging; boosting; classification accuracy.

1. INTRODUCTION

In today's world, machine learning has gained popularity to a great extent. One of the popular mechanisms within supervised learning related to decision tree is random forest. The general method for random forest was first suggested by Ho in 1995. Random forest is ensemble of pruned binary decision tree, unlike other it generates numerous trees which creates forest like classification [1]. Ensemble learning method of the random forest is very promising technique in terms of accuracy [3]. Random forest is one of best techniques used for the classification of unbalanced data in machine learning and data mining for data analysis and data extraction. Random forest has found its wide spread use in various applications [2]. The acceptability of random forest can be primarily attributed to its capability of efficiently handling non-linear classification task. Random forest is well known for taking care of data imbalances in different classes, especially for large datasets [4]. Due to its parallel architecture, random forest classifier is faster compared to other classifiers like ID3, C4.5, and CART [15]. A large number of variants of random forest can be found in literature [5].

According to classification, performance of random forest improves with increase in the number of trees [14]. But experimental evidences suggest that adding trees beyond certain pre-determined limit may not significantly improve the classification performance of random forest [6]. Further, it has been shown that random forest may perform biased feature selection for individual trees [7]. As a result, an unimportant feature may be favored in a noisy feature set. Consequently, classification accuracy may degrade [8]. So, an increased proportion of important features (i.e. removal of unimportant features) may have significant impact on the classification performance of random forest. A number of feature selection strategies can be found in the literature [9]. But features of initial forest leads to the performance criteria.

The rest of the paper is organized into three sections. In Section 2, the methods of feature selection are discussed. Section 3 illustrates briefly the Improved Random Forest. Section 4 gives conclusions and future scope.

2. MECHANISM OF FEATURE SELECTION

2.1. *Feature Selection*

Feature selection is an important preprocessing step in machine learning applications, where it is often used to find the smallest subset of features that maximally increases the performance of the model. Besides maximizing model performance, other benefits of applying feature selection include the ability to build simpler and faster models using only a subset of all features, as well as gaining a better understanding of the processes described by the data, by focusing on a selected subset of features [9].Sampling of forest is done with the help of bagging and boosting where bagging is used to reduce the variance of tree by creating several subsets of data from training sample chosen randomly with replacement. The main goal of bagging is to solve the accuracy of prediction [25]. Boosting coverts the weak learners to strong learners. Boosting is used to create a collection of predictors. The main goal of boosting is to solve net errors from the prior trees [24].

Feature selection algorithms may be divided into filters, wrappers and embedded approaches [21].Filter

methods evaluate quality of selected features, independent from the classification algorithms, while wrapper methods application of a classifier to evaluate this quality. Embedded methods perform feature selection during learning of optimal parameters for example, neural network weights between the input and hidden layer [26].

Feature selection techniques can be divided into three categories, namely feature ranking, finding important and unimportant features, and finding number of trees to be added depending on how they interact with the classifier. Feature selection methods directly operate on the dataset, and provide a feature weighting leading to ranking as output [10]. These methods have the advantage of being fast and independent of the classification model, but at the cost of inferior results. In the next few subsections we discuss our feature selection methods in details with three major steps: feature ranking, finding important and unimportant features and finding the number of trees to be added. The feature ranking which is discussed next.

2.1.1. *Feature Ranking*

In feature ranking, first the calculation of weight is carried out for different features. Then rank those features according to their weight. The features with weight below the threshold are subsequently removed. A feature with higher value weight is taken as important feature for classification. Based on the global weights, ranking of the features is done to find the important and unimportant features in the next section.

2.1.2. *Finding Important and Unimportant Features*

The main purpose is to find out the important and unimportant features from the feature vector (weight). It is not known which and how many features are important. So take a novel strategy to find the important features. Initially, from the ranked list mark some features as 'important' based on a feature weight. Note that, once a feature is marked to be important at a construction pass, it will remain important till the end and will not be removed in the subsequent passes. Thus the probability of discarding an important feature is reduced. Importance of a feature may not be evident immediately at a split. So we do not discard any feature at individual nodes during the growth of a tree. Thus the probability of discarding an important feature is further reduced. After getting certain important and unimportant features, formulate a theoretical bound of maximum number of trees to be added to the forest at that construction pass.

2.1.3. *Finding the number of trees to be added*

To find the number of trees to be added, first define two quantities that controls the classification performance of random forest. These two quantities are strength and correlation. Classification accuracy is defined based on strength and correlation [26]. Find the number of trees to be added using the formulation of classification accuracy.

(1) Probability of Good Split: Probability of good split is the probability that a node is split by an important feature. A good split creates child nodes with more homogeneity compared to the parent node. A good split in node is possible only if at least one important feature is present in corresponding. There is possibility that some feature, present in the bag of unimportant features might turn out to be important in the subsequent construction passes. Hence, if the features selected from only the important features bag, it will lead to greedy selection and which will miss some potential important features [22]. Hence, choose the features from both the bags of important and unimportant features.

(2) Strength: The strength of a forest is dependent on the minimum classification accuracy of individual trees. Hence, define the strength of the forest as the probability that all the nodes in at least one tree has good splits.

(3) Correlation: After probability and strength of forest, correlation between any two trees is a measure of similarity between the trees. For random forest, correlation between trees is dependent on the features used at different nodes of those trees.

3. IMPROVED RANDOM FOREST

Improved Random Forest (IRF) is introduced which takes care of feature selection and sampling by finding optimal number of trees simultaneously. IRF starts with a forest of small number of trees. The initial forest finds a small number of important features. Then at each construction pass, update the list of important and unimportant features through following four steps. First calculate the weights of different features and rank the features based on their weights. Then calculate a threshold weight. The features with weight below the threshold are subsequently removed. Next, from the set of remaining features, mark some features as 'important' based on a novel criterion. The rest of the features are marked as 'unimportant'. Note that, once a feature is marked to be important at a construction pass, it will remain important till the end and it will not be removed in the subsequent passes. After getting certain important and unimportant features, formulate a theoretical bound of maximum number of trees to be added to the forest at that construction pass.

Show that if trees are added satisfying the bound, the classification accuracy of the forest certainly improves. The construction passes are continued until a novel termination criterion is reached. The pipeline of this method is presented in Figure 1.

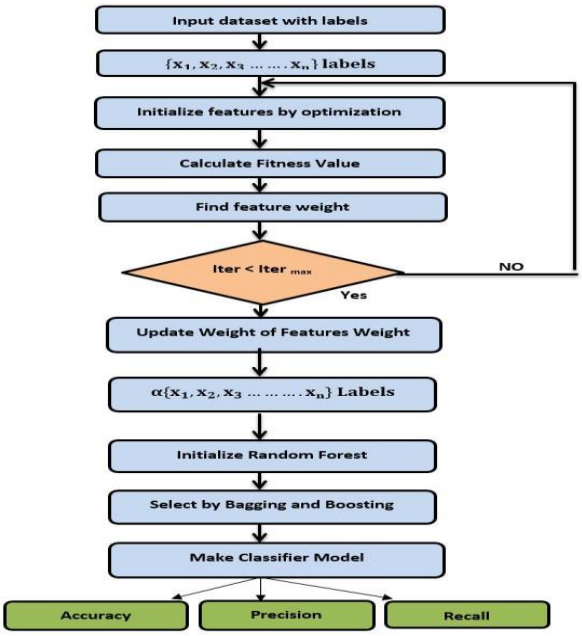

Figure1: The pipeline of the proposed method.

As a result, probability of discarding an important feature is reduced. Now select bagging (bag of important and unimportant features) and boosting (set of predictors to remove error rate) for selecting effective trees for the forest. Thus the proposed forest provides optimal classification accuracy with precision and recall in terms of the number of trees and in terms of feature reduction. Notably, the number of trees in our method is not pre-determined like for specific datasets. So IRF has low data dependence. IRF is fast and hence useful for industrial applications.

3.1.1. *Algorithm*

1. Procedure
2. Initialize forest.
3. Grow initial forest with random trees and feature vector.
4. Calculate weight and rank all the features in using feature vector.
5. From the ranked list of features, sort rank wise.
6. n is the number of construction pass. Initialize n = 0.
7. End procedure
8. While number of unimportant features at the n^{th} construction pass is greater than equal to the number of features from which one is selected for node splitting.

Do

9. Compute mean and standard deviation of feature weights in bag of important features at the n^{th} construction pass.
10. Find features to be removed at the n^{th} construction pass.
11. From the bag of unimportant features, find the set of features whose weights are larger than the minimum of the weights of important features.
12. Find Feature vector at the (n +1) construction pass.
13. Find Bag of important features at the (n+1) construction pass.
14. Find Number of trees at (n+1) construction pass.
15. Select boosting algorithm for the precision and recall. Grow forest using new trees with important features and feature vector.
16. Make Classification Model.
17. End while

3.1.2. *Comparison*

Improved Random Forest compared to conventional Random Forest: IRF is a modification of conventional random forest. It has already been observed that given the same number of trees, IRF outperforms RF. Next we investigate if increasing the number of trees in RF can lead to results comparable to IRF. For each data, we find the numbers of trees in RF that produce the lowest average error by using the algorithm. Even with much larger number of trees, RF does not beat the proposed IRF. Thus we find that our method beats RF with less computational burden. Therefore IRF has low data dependence. IRF is fast and hence useful for industrial applications.

4. CONCLUSIONS AND FUTURE SCOPE

We proposed a fast and accurate solution for automatic classification by improvising random forest classifier. The proposed classifier not only removes redundant features, but also dynamically change the size of the forest (number of trees) to produce optimal performance in terms of classification accuracy. The proposed classifier has the potential to be applied in industrial applications. Future work may include revalidation of Improved Random Forest using any tool on different datasets.

REFERENCES

[1] Miroslav Kubat, "An Introduction to Machine Learning", 2nd edition, Springer International Publishing AG, 2017.
[2] "Introduction to boosting algorithms" from https: //www.analyticsvidhya.com/blog/2015/11/quick-introduction-boosting-algorithms-machine-learning/accessed on 17/01/2018 at 1300 hrs.

[3] "Random Forest" from https://en.Wikipedia.org /wiki /Random_forest accessed on 28/02/ 2018 at 2600 hrs.
[4] "Bagging and random forest ensemble algorithms-for machine learning" from https:// algorithm–machine-learning– mastery.com/bagging-and–random–forest– ensemble–algorithms–for machine - learning/accessed on 23/02/2018 at 2100 hrs.
[5] "Random-Forest-Algorithms" from https: // www .datasciencecentral.com/profiles/blogs/randomforestsalgorithm accessed on 29 /03/2018 at 1200 hrs.
[6] Lomax. S, "A survey of cost-sensitive decision tree induction algorithms", ACM Computing Surveys, pp. 34-44, 2013.
[7] C. Luo, Z. Wang, S. Wang, J. Zhang, and J. Yu, "Locating facial landmarks using probabilistic random forest," Signal Processing Letters, IEEE, vol. 22, no. 12, pp. 2324–2328, Dec 2015.
[8] A. Criminisi and J. Shotton, "Decision forests for computer vision and medical image analysis", Springer Science & Business Media, 2013.
[9] Janecek, "On the Relationship between Feature Selection and Classification Accuracy", JMLR: Workshop and Conference Proceedings, vol. 4, pp. 90-105, 2008.
[10] Girish ChandraShekar, "A Survey on Feature Selection Methods", ELSEVIER, Computer and Electrical Engineering, pp. 16-24, 2014.
[11] T. M. Khoshgoftaar, M. Golawala, and J. V. Hulse, "An empirical study of learning from imbalanced data using random forest," in ICTAI 2007, vol. 2. IEEE, 2007, pp. 310–317.
[12] X.-Y. Liu, J. Wu, and Z.-H. Zhou, "Exploratory undersampling for class-imbalance learning," Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, vol. 39, no. 2, pp. 539–550, 2009.
[13] T. G. Dietterich, "Ensemble methods in machine learning," in multiple classifier systems. Springer, 2000, pp. 1–15.
[14] N. Quadrianto and Z. Ghahramani, "A very simple safe-bayesian random forest," PAMI, IEEE Trans. on, vol. 37, no. 6, pp. 1297–1303, June 2015.
[15] A. Paul, A. Dey, D. P. Mukherjee, J. Sivaswamy, and V. Tourani, "Regenerative random forest with automatic feature selection to detect mitosis in histopathological breast cancer images," in MICCAI 2015. Springer, 2015, pp. 94–102.
[16] L. Breiman, "Random forests," Machine learning, vol. 45, no. 1, pp. 5–32, 2001.
[17] T. M. Oshiro, P. S. Perez, and J. A. Baranauskas, "How many trees in a random forest?" in MLDM. Springer, 2012, pp. 154–168.
[18] P. Latinne, O. Debeir, and C. Decaestecker, "Limiting the number of trees in random forests," in Multiple Classifier Systems. Springer, 2001, pp. 178–187.
[19] A. Cuzzocrea, S. L. Francis, and M. M. Gaber, "An information theoretic approach for setting the optimal number of decision trees in random forests," in Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on. IEEE, 2013, pp. 1013–1019.
[20] C. Strobl, A.-L. Boulesteix, A. Zeileis, and T. Hothorn, "Bias in random forest variable importance measures: Illustrations, sources and a solution," BMC bioinformatics, vol. 8, no. 1, p. 25, 2007.
[21] I. Guyon, S. R. Gunn, A. Ben-Hur, and G. Dror, "Result analysis of the nips 2003 feature selection challenge." in NIPS, vol. 4, 2004, pp. 545–552.
[22] T. Bylander, "Estimating generalization error on two-class datasets using out-of-bag estimates," Machine Learning, vol. 48, no. 1-3, pp. 287–297, 2002.
[23] H. Ishwaran, "The effect of splitting on random forests," Machine Learning, vol. 99, no. 1, pp. 75–118, 2014.
[24] Y. Freund, R. E. Schapire, "Experiments with a new boosting algorithm," in ICML, vol. 96, 1996, pp. 148–156.
[25] C. Seiffert, "Rusboost: A hybrid approach to alleviating class imbalance," SMC, Part A: Systems and Humans, IEEE Trans. on, vol. 40, no. 1, pp. 185–197, 2010.
[26] Eugene Tuv., "Feature Selection with Ensembles, Artificial Variables, and Redundancy Elimination", Journal of Machine Learning Research, vol. 10, pp. 1341-1366, 2009.