Exponentially Weighted Random Forest

Vikas Jain, Jaya Sharma, Kriti Singhal, and Ashish Phophalia(✉)

Indian Institute of Information Technology, Vadodara, Gandhinagar, India
{201671001,201551021,201551024,ashish_p}@iiitvadodara.ac.in

Abstract. Random forest (RF) is a supervised, non-parametric, ensemble-based machine learning method used for classification and regression task. It is easy in terms of implementation and scalable, hence attracting many researchers. Being an ensemble-based method, it considers equal weights/votes to all atomic units i.e. decision trees. However, this may not be true always for varying test cases. Hence, the correlation between decision tree and data samples are explored in the recent past to take care of such issues. In this paper, a dynamic weighing scheme is proposed between test samples and decision tree in RF. The correlation is defined in terms of similarity between the test case and the decision tree using exponential distribution. Hence, the proposed method named as Exponentially Weighted Random Forest (EWRF). The performance of the proposed method is rigorously tested over benchmark datasets from the UCI repository for both classification and regression tasks.

Keywords: Classification · Ensemble method · Random forest

1 Introduction

Random forest (RF) is an ensemble-based, supervised machine learning algorithm proposed by Leo Brieman [6][1]. It consists of numerous randomized decision trees to solve classification and regression problems. In RF, decision trees are constructed independently. Therefore, RF can be implemented and executed as parallel threads, hence it is fast and easy to implement. It has been used for various domains like brain tumor segmentation, Alzheimer detection, face recognition, human pose detection, object detection etc [7].

A decision tree in RF is built during the training phase using the bagging concept. A decision tree has several important parameters like predefined splitting criteria, tree depth and the number of elements on the leaf node. However, the best choice of these parameters is not answered precisely yet [7,10]. This motivated various methods to come up with the heuristic approach in building the decision tree and hence RF. The method proposed by Paul et al. [15] converges with reduced and important features, and derived the bound for the

[1] Referred to as conventional random forest throughout the text.

J. Sharma and K. Singhal—Equal contribution.

B. Deka et al. (Eds.): PReMI 2019, LNCS 11941, pp. 170–178, 2019.
https://doi.org/10.1007/978-3-030-34869-4_19

number of trees. In addition, there has been some work done on proving the consistency of RF and leveraging dependency on the data by several researchers [4,5,9,16]. Denil et. al. [9] used Poisson distribution in feature selection for growing a tree, whereas Wang et al. [16], has proposed a Bernoulli Random Forest (BRF) framework incorporating Bernoulli distribution for the feature and splitting point selection.

The conventional RF assigns equal weights to the votes casted by each individual tree [6]. Hence, the prediction is made based on the majority voting. However, in the real-life scenario, a dataset may have a huge number of features, but the percentage of truly informative features may be less. Therefore, the contribution of such decision trees, which are populated by less informative attributes may be less. Hence, all the trees in a forest are not equally contributing to the better classification [8]. Therefore, instead of assigning a fixed weight to the decision tree, the dynamic weight should be assigned. Paul et al. [13] have proposed a method to compute the weights during the training phase and assigns a fixed weight to each decision tree. The mechanism proposed by Winham et al. [17] and Liu et al. [12], both computes the weight either based on the performance of tree computed using OOB samples or using a feature weighing scheme. Akash et al. [2] compute the confidence as weight in RF using the entropy or Gini score calculated during the tree construction. However, these methods do not talk about the relationship of these weights with test samples. Therefore, a dynamic weighing scheme is proposed in this paper. It computes the similarity between test cases and the decision tree using exponential distribution. Therefore, the forest formed is named as Exponentially Weighted Random Forest (EWRF).

The remainder of this paper is organized as follows: Sect. 2, describes RF as a classifier and regression and problem associated with conventional RF. Section 3, presents the proposed EWRF approach. Section 4, discuss the implementation details and performance. It has been concluded in Sect. 5.

2 Random Forest

Random forest built upon decision trees as an atomic units. Each decision tree either behaves as a classifier for classification or as a regressor to predict the output for regression task. Given a dataset $\mathbb{D} = \{(X_1, C_1), (X_2, C_2), ..., (X_M, C_M)\}$ with M number of instances such that $X_i \in \mathbb{R}^N$ with N number of attributes. Let the dataset is having class labels as $C_i \in \{Y_1, Y_2, ..., Y_C\}$. Initially, dataset $\mathbb{D}$, is partitioned into training set $\mathbb{D}_1$, having M' number of instances ($M' < M$), and testing set $\mathbb{D}_2$, having remaining instances. Decision trees are constructed using training samples along with bootstrap sampling (random sampling along with replacement) as described in [6].

2.1 Random Forest as Classifier

Random forest assigns the class value based on the proportion of the individual class values present at the leaf node.

The class distribution for the j^{th} class at the terminal node h, in the decision tree t, for the test case X, can be represented as:

$$p_{j,h}^{t} = \frac{1}{n_h} \sum_{i \in h} \mathbb{I}(Y_i = j) \tag{1}$$

here: n_h is total number of instances in the terminal node h. $\mathbb{I}(\cdot)$ is an Indicator function.

Based on maximum class distribution, the class value j, is assigned by the decision tree t, for the test case X, by the following equation:

$$\hat{Y}_j^t = \max_{1 \le j \le C} \{p_{j,h}^t\} \tag{2}$$

To assign the final class value based on majority voting in conventional RF, first count the predicted class by each decision tree for the test case X, using the following equation:

$$C(Y_i = j) = \sum_{t=1}^{n_{tree}} \mathbb{1} \cdot \mathbb{J}(\hat{Y}_j^t) \tag{3}$$

here, $\mathbb{J}(\cdot)$ is an indicator function. Finally, based on majority voting, RF assigns the final class value using Eq. (4).

$$\hat{Y} = \max_{1 \le j \le C} \{C(Y_i = j)\} \tag{4}$$

2.2 Random Forest as Regressor

In regressor task, decision trees have to predict the outcome. In the regression dataset, the outcome value associated with each instance is a single real value i.e. $\mathbf{Y_i} \in \mathbf{R}$. In order to construct RF as a regressor, Mean Squared Error (MSE) is used as the splitting criterion. Once all the decision trees are constructed, the test instance is passed to each decision tree. Based on the decision tree node values, test instance follows either left or right subtree and reaches to the leaf node. The predicted value is the mean value of instances present at the leaf node. The predicted value for a test case $\mathbf{X}$, at a terminal node h, by the decision tree t, is the mean value of instances present within the leaf node. It can be calculated as:

$$\hat{Y}_h^t = \frac{1}{n_h} \sum_{y_i \in h} Y_i \tag{5}$$

Finally the predicted value by the RF is the average of values predicted by each trees. Hence, the overall prediction made by forest can be computed as:

$$\hat{Y} = \frac{1}{n_{tree}} \sum_{t=1}^{n_{tree}} \mathbb{1} \cdot \hat{Y}_h^t \tag{6}$$

2.3 Problem with Conventional Random Forest

Random forest classifier to be effective, each decision tree must have reasonably good classification performance and trees must be diverse and weakly correlated [14]. The diversity is obtained by randomly choosing training instances and attributes for each tree. However, a decision tree can not always contribute effectively to each and every test instance. Considering a dataset with a high ratio of less informative attributes, the performance of RF gets significantly affected. This is due to the equal contribution of decision trees while performing majority voting. In such cases, performance can be increased by reducing the contribution of decision trees whose nodes are populated by non-informative attributes and assigning a dynamic weight to the decision trees [3,11].

3 Proposed Method

The proposed EWRF consists of two steps. In the first step, decision trees are constructed as described in conventional RF [6]. In the second step, the exponential weight score is calculated as described in following subsections.

3.1 Exponential Weight Score Calculation

During the testing phase, test samples are passed to each and every decision tree in the forest. Let F_i is the feature value for splitting at an internal node of a decision tree t. A test sample $X = \{a_1, a_2, ..., a_j, ..., a_N\}$, is passed to a decision tree. It is guided either to the left ($a_j^X \leq \tau$) or right ($a_j^X > \tau$) subtree, based on threshold τ, and move down until it reaches to the leaf node of decision tree t. The sum of the squared distance between corresponding attribute values in the test sample X, and participating nodes F_i, in the path of the decision tree t, is calculated as follows:

$$d = \sum ||F_i - a_j^X||_2; \forall F_i \in t; a_j \in X$$

Thus, we have $\{d_1, d_2, ..., d_{n_{tree}}\}$ distances computed for each test sample, with respect to all decision trees. The smaller the value of d for the decision tree, the more will be the similarity between tree and test case till that node, and hence the corresponding will be high weight value. This has been shown in Fig. 1. In the proposed EWRF, the weight associated with each decision tree directly proportional to the similarity between the test instance and decision tree. Hence, the weight associated with a decision tree is computed using an exponential distribution measure to maintain such a relationship. In this way, the weight of each decision tree for incoming test cases may vary. The exponential tree weight score is calculated as follows:

$$W_{\mathbf{X}}^t = \frac{1}{Z} \exp\left\{-\frac{\sum ||F_i - a_j^X||_2}{\alpha}\right\} \tag{7}$$

Algorithm 1. Prediction(X)

Input: n_{tree} = # trees, and test case X_i
Output: Predicted class / output value
for *tree t,* **to** n_{tree} **do**
- Calculate the sum of difference of distance d, between the attribute value of test case and corresponding attributes values of participating nodes of decision tree in the path followed by test case ;
- Calculate the Exponential weight score for each tree t, using Equation (7) and store it into a list ;
- For classification, store the class value with maximum proportion, refer Equation (2) OR
For regression, store the predicted value as mean of instances present at leaf node, refer Equation (6) ;

end
- Normalize the weight score calculated for each tree t, and assign this value to the concerned tree ;
- Multiply the Exponential weight score to each decision tree and perform majority voting, refer equation (8) and (9);
- Return:
The class value, for Classification
OR
The predicted output value, for Regression ;

where Z is the normalizing term, which is the sum of weights of all dsecision trees. The α value is one of the hyper-parameter to control the weight score. For classification, the Eq. (3) is turned out to be as:

$$C(Y = j) = \sum_{t=1}^{n_{tree}} (W_{\mathbf{X}}^{t}) \cdot J(\hat{Y}_{j}^{t}) \tag{8}$$

For regression, the Eq. (6) is turned out to be as:

$$\hat{Y} = \frac{1}{n_{tree}} \sum_{t=1}^{n_{tree}} (W_{\mathbf{X}}^{t}) \cdot \hat{Y}_{h}^{t} \tag{9}$$

At last, weighted voting is performed using Eqs. (8) and (9) for predicting output in classification and regression tasks respectively, shown in Fig. 2. The pseudo code for predicting the class or regression value is provided in Algorithm 1.

4 Experimental Results

This section is comprised of datasets, implementation details, and performance analysis of EWRF compared to conventional RF, and state-of-the-art methods.

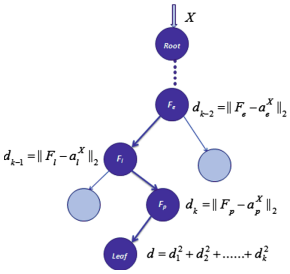

Fig. 1. An example to show the calculation of distance during testing. In this example an test instance X follows the path marked as bold blue lines (F_e, F_l and F_p) to reach up to leaf node. The distance is calculated at the corresponding node in the path followed by the test case. At the root node, all distances are sum up to get the final distance between test case and decision tree. (Color figure online)

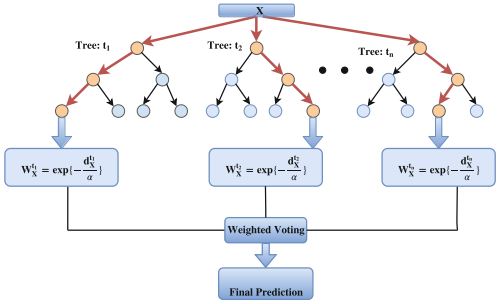

Fig. 2. The proposed EWRF method to show how exponentially weighted score is calculated by different decision trees, for the given test instance. Further, weighted voting is performed for final prediction

4.1 Datasets and Implementation Details

The experiments have been conducted over the benchmark datasets, which are publicly available over the UCI repository [1]. These datasets are from a variety of domains and have different combinations of numerical attribute values. These datasets vary in terms of the number of classes, features, and instances for rigorous testing of the proposed method.

There are five main parameters for conducting the experiments: (1) the number of trees n_{tree}, (2) the number of minimum instances at leaf node n_{min}, (3) the sample ratio in which dataset is divided into training set and test set, (4) the maximum tree depth T_{depth}, and (5) value of α for computation of exponential weighing score. The value of n_{tree} is decided empirically. The experiments have been done over Vehicle, Wine, and Abalone datasets with n_{tree} in the range of 10 to 100 with a step size of 10. We have observed that beyond $n_{tree} = 50$, the accuracy saturates, so it is kept as 50 in all experiments. The n_{min} is kept as 5 and the sample ratio for dividing the datasets into training and testing is kept as 0.5. These values are taken from the state-of-the-art methods for a fair comparison. Experiments have been done with different values of T_{depth} and the results are quoted with the depth, where accuracy is better among different trials. The value of α is chosen as 0.45 for classification and 0.75 for regression. It is also decided by experimenting with different values of $\alpha = \{0.15, 0.45, 0.75, 1.0\}$. Each of the experiment is repeated 10 times with the random selection of training and testing subsets.

Table 1. MSE comparison between state-of-the-art methods and proposed EWRF with average over 10 iterations (least value is the best)

SN	Dataset	Dimension	RF	Biau08	Biau12	Denil	BRF	EWRF
1	Slump	103*10	**41.58**	62.30	62.31	60.19	55.67	45.63
2	Servo	167*4	2.58	3.19	2.39	2.26	2.07	**0.91**
3	Automobile	205*26	1.62	1.53	1.51	1.46	1.41	**1.18**
4	Yacht	308*7	50.6	229.86	225.97	150.58	128.85	**35.88**
5	Housing	506*14	27.7	85.50	82.97	81.62	77.81	**27.1**
6	Student	649*33	41.9	9.83	9.81	9.38	8.93	**4.1**
7	Concrete	1030*9	130.5	279.13	279.70	278.64	275.56	**125.54**
8	Wine quality	4898*12	0.57	0.81	0.81	0.67	**0.51**	0.57
9	Airfoil	1503*6	28.9	66.66	47.73	43.47	38.57	**26.2**
10	Energy_y1	768*8	**2.33**	64.11	40.71	24.53	19.85	5.1

Table 2. Classification accuracy comparison between state-of-the-art methods and proposed EWRF with average over 10 iterations (high value is the best)

SN	Dataset	Dimension	# Classes	RF	Biau08	Biau12	Denil	BRF	EWRF
1	Transfusion	748*5	2	72.2	68.92	70.27	72.97	**77.7**	72.9
2	Spambase	4601*57	2	91.1	60.59	60.59	**94.4**	94.1	90.7
3	CVR	435*16	2	88.8	51.86	61.4	94.4	**95.6**	94.4
4	Madelon	2600*500	2	60.6	49.27	50.31	54.81	**69.2**	58.8
5	Wine	178*13	3	96.9	40.59	41.18	96.47	97.7	**98.3**
6	CMC	1473*9	3	50.29	42.72	42.65	53.6	54.6	**55.2**
7	Verbetral	310*6	3	80.9	48.39	48.39	82.26	82.3	**82.9**
8	Connect-4	67557*42	3	64.58	64.52	65.47	66.19	76.19	**77.1**
9	Vehicle	946*18	4	72.6	27.98	23.1	68.81	71.67	**73.5**
10	Zoo	101*17	7	85.3	50	41	80	85	**87.2**
11	Abalone	4177*8	29	**27.1**	16.05	16.52	26.23	26.44	**27.1**

4.2 Performance Analysis

The results generated with the proposed EWRF are compared to the conventional RF [6], and the state-of-the-art methods, i.e. four variants of random forest Biau08 [5], Biau12 [4], Denil [9] and BRF [16] for the regression and classification datasets. The highest learning performance among these comparisons is marked in boldface for each dataset.

In regression, it can be observed from Table 1 that EWRF achieves the significant reduction in MSE on seven datasets out of ten datasets. In particular, one can observe that for the Concrete dataset, Biau08 [5], Biau12 [4], Denil [9], and BRF [16] have almost same MSE value. However, there is more than 50% reduction in MSE for Yacht, Concrete, and Housing datasets. The proposed method has also shown improvement for datasets having large number of classes like

Student, and Automobile. From Table 1, it is clear that the proposed method has shown much improvement over the compared state-of-the-art methods.

For classification, the comparison between the existing state-of-the-art methods and proposed EWRF is shown in Table 2. It can be seen that EWRF is showing improvement as compare to Biau08 [5], Biau12 [4] and Denil [9] for all the classification data except for Spambase. In comparison with BRF [16], the proposed method is showing improvement for seven datasets out of eleven datasets. In comparison to conventional RF, the proposed EWRF is showing improvement for nine datasets out of eleven datasets.

5 Conclusion

The conventional Random Forest (RF) assigns equal weights to the votes cast by each individual tree. Also, the approaches proposed in the past assigns weights to every decision tree during the training phase only. In this paper, we have explored the dynamic relationship between test samples and decision trees, based on which aggregation/weighted voting is performed. Thus, weights derived in EWRF are dynamic in nature. The proposed method is tested over various heterogeneous datasets and compared to state-of-the-art competitors. The proposed method has shown improvement for both regression and classification tasks.

References

1. UCI repository. https://archive.ics.uci.edu/ml/index.php. Accessed 15 Nov 2018
2. Akash, P.S., Kadir, M.E., Ali, A.A., Tawhid, M.N.A., Shoyaib, M.: Introducing confidence as a weight in random forest. In: 2019 International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST), pp. 611–616. IEEE (2019)
3. Amaratunga, D., Cabrera, J., Lee, Y.S.: Enriched random forests. Bioinformatics **24**(18), 2010–2014 (2008)
4. Biau, G.: Analysis of a random forests model. J. Mach. Learn. Res. **13**(Apr), 1063–1095 (2012)
5. Biau, G., Devroye, L., Lugosi, G.: Consistency of random forests and other averaging classifiers. J. Mach. Learn. Res. **9**(Sep), 2015–2033 (2008)
6. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)
7. Criminisi, A., Shotton, J.: Decision Forests for Computer Vision and Medical Image Analysis. Springer (2013)
8. Deng, H., Runger, G.: Feature selection via regularized trees. In: The 2012 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE (2012)
9. Denil, M., Matheson, D., De Freitas, N.: Narrowing the gap: random forests in theory and in practice. In: International Conference on Machine Learning, pp. 665–673 (2014)
10. Ishwaran, H.: The effect of splitting on random forests. Mach. Learn. **99**(1), 75–118 (2015)
11. Kulkarni, V.Y., Sinha, P.K., Petare, M.C.: Weighted hybrid decision tree model for random forest classifier. J. Inst. Eng. (India): Ser. B **97**(2), 209–217 (2016)

12. Liu, Y., Zhao, H.: Variable importance-weighted random forests. Quant. Biol. **5**(4), 338–351 (2017)
13. Paul, A., Mukherjee, D.P.: Enhanced random forest for mitosis detection. In: Proceedings of the 2014 Indian Conference on Computer Vision Graphics and Image Processing, p. 85. ACM (2014)
14. Paul, A., Mukherjee, D.P.: Reinforced random forest. In: Proceedings of the Tenth Indian Conference on Computer Vision, Graphics and Image Processing, p. 1. ACM (2016)
15. Paul, A., Mukherjee, D.P., Das, P., Gangopadhyay, A., Chintha, A.R., Kundu, S.: Improved random forest for classification. IEEE Trans. Image Process. **27**(8), 4012–4024 (2018)
16. Wang, Y., Xia, S.T., Tang, Q., Wu, J., Zhu, X.: A novel consistent random forest framework: Bernoulli random forests. IEEE Trans. Neural Netw. Learn. Syst. **29**(8), 3510–3523 (2018)
17. Winham, S.J., Freimuth, R.R., Biernacka, J.M.: A weighted random forests approach to improve predictive performance. Stat. Anal. Data Min.: ASA Data Sci. J. **6**(6), 496–505 (2013)