International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 04, April-2021

Prediction of Road Traffic Concentration using Random Forest Algorithm based on Feature Compatibility

Ayana Aboma Regassa
School of Information Technology and Engineering
Tianjin University of Technology and Education
Tianjin, China

Wu Zhi Feng
School of Information Technology and Engineering
Tianjin University of Technology and Education
Tianjin, China

***Abstract*—Different algorithms of decision tree are commonly used as the base classifiers of random forest algorithm. In order to solve in this paper. In random forest algorithm based of feature compatibility, the problem that the classifier is biased to select redundant features and contains a lot of feature space is addressed. Considering the micro logical relationship and coordination correlation between features, feature compatibility of random forest is introduced. This proposed algorithm mainly uses feature ranking that includes feature selection for easing the number of input variables, which in turn useful in reducing computational cost and in moderate number of features improves the performance of the model .These ranking is based on, features with higher value weight is more important for classification and regression. After ranking correlation considers initial feature vector, entropy based measure for node splitting, the probability of class at that node and entropy of the node. Using correlation, we are able to identify the degree of coherence between each. Features are positively or negatively correlated between themselves and between the targets. By using the features that are not well correlated between themselves the feature with the largest negative correlation with other is selected for regression attribute to be used in the data set, then apply it. In this paper, extremely random forest is also introduced and implemented. Extremely random forests take randomness to the next level. Along with taking a random subset of features, the thresholds are chosen randomly as well. These randomly generated thresholds are chosen as the splitting rules, which reduce the variance of the model even further. Outliers are the values that escapes normality and probably cause anomalies in the results obtained through algorithms and analytical systems.The availability of outliers and the way to deal them is applied in the paper. Better evaluation methods of the models using cross validation is also applied. Lastly, UCI data set is used to verify the accuracy of the algorithms. The proposed algorithm has higher accuracy with average amount of attributes than traditional random forest algorithm and extremely random forest algorithm but higher training accuracy with equal number of attributes with random forest algorithm. Additionally the proposed algorithm has the overall shortest running time.**

***Keywords*—Random Forest; Extremely Random Forest;Feature Compatibility ; Outliers ; cross-validation**

I. INTRODUCTION

Machine learning is about making machines get better at some task by learning from data, instead of having to explicitly code rules. Data is gathered into a training set, and fed the training set to a random forest, extremely random forest and random forest based on feature compatibility algorithm. the algorithm is model-based it tunes some parameters to fit the model to the training, set that is to make good predictions on the training set itself, and this hopefully will be able to make good predictions on new cases as well. For instance based algorithms, it just learns the examples by and uses a similarity measure to generalize new instances.

The system will not perform well if the training set is too small, or if the data is not representative, noisy, or polluted with irrelevant features. The models needs to be neither too simple in which case it will underfit nor too complex in which case it will overfit. Random forest has found its wide spread acceptability in various applications [1],[2],[5]. The acceptability of random forest can be primarily because of its capability to efficiently handle non-linear classification task. Random forest is well known for handling of data imbalances in different classes [6], [7] especially for large datasets [8].

In order to overcome the problem that base classifiers are easy to fall into local optimal solutions and over-fitting, Leo Breiman proposed a random forest algorithm [4], which has lower generalization error [4] and better convergence [4].We know that a forest is made up of trees and more trees means more robust forest. Similarly, the algorithms creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting.

The method considers the correlation between the condition feature and the label feature, and has certain improvement on the performance of the classifier. Based on the degree of correlation, we consider the logical relationship between the features, introduce the concept of feature compatibility [9], and use the negative correlation of features as the standard for selecting them for the classifier.

II. FUNDAMENTALS OF RANDOM FOREST ALGORITHM

A. Construction of random forests

The algorithm for building a random forest is as follows:

(I) Subset the original data so that the decision tree is built on only a sample of the original dataset.

(II) Subset the independent variables or features too while building the decision tree.The first and second step mainly

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 04, April-2021

use Bootstrapping aggregating [10] sampling technique in the random forest algorithm to generate k training subsets with certain repetitions from the original training set through random and then put back sampling methods.
(III) Build a decision tree based on the subset data where the subset of rows as well as columns is used as the dataset.
After the training subsets are obtained, the feature sub spaces [11] (the number of features is usually $[\log_2 M] + 1$, M is the total number of features) are selected from each training subset to generate the k decision trees, thereby forming a "random forest". Each decision tree is allowed to grow without repotting [12].

(IV) Predict on the test or validation dataset.

(V) Repeat steps I through III k number of times, where k is the number of trees built.
(VI) The final prediction on the test dataset is the average of predictions of all k trees.

B. Random Forest Dictum

(1) Random Forests for Regression

Random forests for regression are formed by growing trees depending on a random vector Θ such that the tree predictor h(x,Θ) takes on numerical values as opposed to class labels. The output values are numerical and we assume that the training set is independently drawn from the distribution of the random vector Y,X. The mean-squared generalization error for any numerical predictor h(x) is

$$E_{X,Y}{}^{(Y-h(X))^2} \quad (1)$$

The random forest predictor is formed by taking the average over k of the trees {h(x,Θk)}. Similarly, to the classification case, the following holds: As the number of trees in the forest goes to infinity, almost surely,

$$E_{X,Y}{}^{(Y-av_k h(X,\Theta_k))^2} \rightarrow E_{X,Y}{}^{(Y-E_\Theta h(X,\Theta))^2} \quad (2)$$

Assume that for all Θ, $EY=E_X h(X,\Theta)$. Then

PE(forest) ≤ $\bar{\rho}$PE*(tree) where*

PE*(tree) $=E_\Theta E_{X,Y}{}^{(Y-h(X,\Theta))^2}$, *$\bar{\rho}$ is the weighted correlation between the residuals Y−h(*X,Θ*) and Y−h(*X,Θ'*) where Θ,Θ' are independent*

(2) Empirical Results in Regression

In regression forests, we use random feature selection on top of bagging. Therefore, we can use the monitoring provided by out-of-bag estimation to give estimates of PE*(forest), PE*(tree) and $\bar{\rho}$. These are derived similarly to the estimates in classification. Throughout, features formed by a random linear sum of two inputs are used. We comment later on how many of these features to use to determine the split at each node. The more features used, the lower PE*(tree) but the higher $\bar{\rho}$.

(3) Select a performance measure

In this part our task is to select a performance measure. A typical performance measure for regression and prediction problems is the Root Mean Square Error (RMSE), along with Mean Square Error (MSE) and Mean Absolute Error (MAE). It measures the standard deviation of the errors the system makes in its predictions.

$$\text{RMSE}(\mathbf{X},h) = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(h(x^i) - y^{(i)})^2} \quad (3)$$

The standard deviation, generally denoted σ (the Greek letter sigma), is the square root of the variance, which is the average of the squared deviation from the mean. Where

- m is the number of instances in the dataset we are measuring the RMSE on.
- $x^{(i)}$ is a vector of all the feature values (excluding the label) of the i^{th} instance in the dataset, and $y^{(i)}$ is its label (the desired output value for that instance).
- X is a matrix containing all the feature values (excluding labels) of all instances in the dataset. There is one row per instance and the i^{th} row is equal to the transpose of $x^{(i)}$, noted $(x^{(i)})^{T.6}$.
- h is our system's prediction function, also called a hypothesis. When our system is given an instance's feature vector $x^{(i)}$, it outputs a predicted value $\hat{y}^{(i)} = h(x^{(i)})$ for that instance (ŷ is pronounced "y-hat").
- RMSE(X,*h*) is the cost function measured on the set of examples using your hypothesis *h*.

We use lowercase italic font for scalar values (such as *m* or $y^{(i)}$) and function names (such as *h*), lowercase bold font for vectors (such as $x^{(i)}$), and uppercase bold font for matrices (such as X).
Even though the RMSE is generally the preferred performance measure for regression tasks, in some contexts we may prefer to use another function. For example, suppose that there are many outlier districts. In that case, we may consider using the Mean Absolute Error (also called the Average Absolute Deviation)

$$\text{MAE}(X,h) = \frac{1}{m_i}\sum_{i=1}^{m}\left|h(x^{(i)}) - y^{(i)}\right| \quad (4)$$

(4) Out of Bag estimation

In this part, using the out-of-bag estimated values for the outputs instead of the actual training set outputs gives more accurate trees. We used simple and accurate out-of-bag estimates that can be given for the generalization error of bagged predictors. Accuracy is increased if the prediction method is unstable, that is if small changes in the training set or in the parameters used in construction can result in large changes in the resulting predictor. Besides its primary purpose of increasing accuracy, has valuable byproducts. Roughly, 37% of the examples in the training set T do not appear in a particular bootstrap training set T_B.Means since

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 04, April-2021

the Bagging method randomly extracts training samples from the original sample *T* each time, about 37% of the samples do not appear in the sampled data T_B, T_B is used as a training set, $T \setminus T_B$ as a test set; So both the evaluation model and the model to be evaluated are used N training samples, while still about 25% of the data did not appear in the training set for testing. Like test results are called out-of-bag estimate. The lower correlation between the features is, higher the performance of the base classifier is [12] and the lower the error bound is. In this paper, we start with improving the performance of the base classifier, reconstruct the partitioning rules of the base classifier, and improve the performance of the base classifier, thereby improving the performance of random forests.

III. ENHANCED ALGORITHM

The main idea of the improved method is to consider the micro-logical relationship between features based on the degree of decision coordination and correlation between them, and define feature compatibility [13]

A. Feature Ranking

Feature selection is the process of reducing the number of input variables when developing a predictive model. It is desirable to reduce the number of input variables to both reduce the computational cost of modeling and, in some cases, to improve the performance of the model.

Feature selection methods involve evaluating the relationship between each input variable and the target variable using different evaluation methods and selecting those input variables that have the strongest relationship with the target variable. These methods are fast and effective, although the choice of statistical measures depends on the data type of both the input and output variables. After feature selection feature ranking follows. Consider an initial feature vector $F_0(.)$. While growing a random tree, we use a entropy based measure for node splitting. Let us take a node i in a tree τ (14).

Let the probability of class label c at this node be p(c).

Then entropy of that node is

$$E = \sum_{\forall C} p(c) \ln \frac{1}{p(c)} \tag{5}$$

For splitting this node, we first choose a set (Λ) of f features randomly from $F_0(.)$ without replacement. Assume that feature *j* is present in Λ and we split node *i* with feature *j*. Let the resultant left and right child of node i have entropy E_l and E_r

respectively. Then, for node i, we define the quality of split by feature j as: *Q(i, j) = exp {− (El + Er)}*. The feature that provides highest quality of split is chosen to split the node (it is called 'split feature'). Let N be the total number of nodes in tree τ. First, a local weight is assigned to feature j with respect to tree τ:

$$\omega^{\tau}(j) = \frac{\sum_{i=1}^{N} Q(i,j)}{N} \tag{6}$$

The higher the value of $\omega^{\tau}(j)$ the better is the quality of split by feature *j* in tree τ. We calculate weights of the trees based on out-of-bag error [15].

Let (δ^{τ}) be the out-of-bag error for tree τ. Then the normalized weight of tree τ is:

$$\gamma^{\tau} = \frac{1/\delta^{\tau}}{\max_{\tau}(1/\delta^{\tau})} \tag{7}$$

Higher value of γ^{τ} indicates less classification error by tree τ. Hence, features used for splitting the nodes of tree τ are more discriminative features. Using the local weight of feature and the weights of the trees, we calculate global weight of feature j:

$$\omega(j) = \frac{\sum_{\forall_{\tau}} \omega^{\tau}(j)\gamma^{\tau}}{\max_{j} \sum_{\forall_{\tau}} \omega^{\tau}(j)\gamma^{\tau}}. \tag{8}$$

A feature with higher value weight is more important for classification. Based on the global weights *w(j),* features are ranked as important and unimportant features. Which and how many features are important are unknown. So a unique strategy is needed to take to find the important features

Initially, from the ranked list, we mark top u_0 features as 'important' and rest of the features as 'unimportant'

Let Γ (initially Γ_0) be the bag of important features and Γ' (initially Γ'_0) be the bag of unimportant features. These bags of features are updated at every construction pass. Consider the n^{th} pass. Let μ_n and σ_n be the mean and standard deviation of the global weights of the features present in the bag of unimportant features Γ'_n.Then we put the features with global weight $<(\mu_n - 2\sigma_n)$ in a set R_n. The features in R_n are discarded.

Now let there be some feature *j* with weight *w(j)* in the bag of unimportant feature Γ'_n. Assume that *w(j)* is greater or equal to minimum of feature weights in Γ_n. Then feature *j* is marked as important. We promote *j* from Γ'_n to Γ_n . Therefore, a feature j in Γ 0 n is newly marked as important if

$$w(j) \geq \min_{k} w(k); j \epsilon \Gamma'_n, k \epsilon \Gamma_n \tag{9}$$

B. Correlations

Next we calculate correlation between any two trees in the forest. Correlation 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 [16]. At any point of time, at most B/2 tree pairs may be found in a forest with B trees. Consider a pair of trees τ1 and τ2. For splitting node i in τ1 first features f is randomly choose in $\Lambda_i \tau 1$.This selection can be made in $\binom{u+v}{f}$ ways. The same case applies for $\Lambda_i \tau 2$ (set of features for node i in tree τ2).

So, the probability *(p')* that at least one feature is common in $\Lambda_i \tau 1$ and $\Lambda_i \tau 2$ is

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 04, April-2021

$$p' = 1 - \frac{\binom{u+v}{f}\binom{u+v-f}{f}}{\binom{u+v}{f}\binom{u+v}{f}} = 1 - \frac{\binom{u+v-f}{f}}{\binom{u+v}{f}} \quad (10)$$

Then the probability (ρ) that all the N_{av} pairs of nodes in tree τ1 and τ2 have at least one feature in common, is given by:

$$p = (p')^{N_{av}} = \left(1 - \frac{\binom{u+v-f}{f}}{\binom{u+v-f}{f}}\right)^{N_{av}} \quad (11)$$

Notice that $p \ll 1, \forall (u, v)$. Hence $\frac{\partial p}{\partial u} \to 0$ and $\frac{\partial p}{\partial v} \to 0$ Therefore, we define correlation (η_c) as the probability that at least one pair of nodes (from two different trees) have at least one feature in common:

$$\eta_c = \sum_{\tau=1}^{B/2} \binom{B/2}{\tau} p^{\tau}(1-p)^{(B/2)-\tau} = 1\text{-}(1-p)^{B/2} \quad (12)$$

After sorting the features based on the importance, the zero importance feature are spotted, those zero importance features (unimportant features) from our dataset will be dropped. Then using data analysis graphs pair plot and correlation heat map, especially correlation heatmap, we can observe the degree of relation between each features except for the target feature (attribute). The correlation between the features is displayed in customizable numbers or colors. So our choosing criteria is selecting the feature with most negative correlation with other features and drop the features that are well correlated between themselves. Because well-correlated features between themselves gives us redundant information and it helps to remove them if we have many features. If there is n_k numbers of features, count the positive, zero and negative correlation n_i has with $n_0, \ldots, n_{k-1}$ then compare the count of each feature . So if the feature n_i has large number of positive correlation than negative correlation, n_i will be dropped .But if n_i has large number of negative correlation than positive correlation then, n_i will be selected.

IV. RESULTS COMPARISON

A. Datasets

We select the dodgers loop sensor UCI public data set to test the performance of the improved algorithm, random forest algorithm and extremely random forest algorithm performance. These dataset provides the number of cars counted by sensor every 5 min over 25 weeks. The sensor was for the Grendale on the ramp for the 101 North Freeway in Los Angeles. The goal of this data was to predict the presence of baseball game at dodgers stadium and the goal of our model is to predict the number of car that passes through that road. We choose the dataset considering the best Case scenario because it is close enough to the stadium to see unusual traffic after a Dodgers game, but not so close and heavily used by game traffic so that the signal for the extra traffic is overly obvious. This is an on ramp near the stadium so event traffic begins at or near the end of the event time. Our model learns from this data and is able to predict the correct decision given all the other metrics.

B. Experimental results and evaluation

We check the accuracy of the random forest, extremely random forest and the improved random forest algorithm using Anaconda Navigator 3, Jupyter notebook software and python programming language. The comparison result is shown in the table below.

Algorithm	No of attributes	r2 score	*Training Score*	*hyper tuned r2 score*
Random Forest	17568	**77.19%**	93.79%	80.76%
Extremely Random Forest	17568	73.65%	**97.07%**	**81.70%**
Random forest based on Feature compatibility	17568	76.78%	94.08%	80.1%

Table 1.Comparison between the traditional

RF algorithm, Extremely RF algorithm and RF algorithm based on feature compatibility.

From the experimental result, it can be seen that with the same amount of data the proposed method has higher training accuracy than the normal random forest algorithm and less training accuracy than extremely random forest algorithm. Additionally the model accuracy (r2 score) of the proposed algorithm is small compared to RF for the same amount of data, but RF based on feature compatibility outperforms both algorithms with training score, model accuracy and hyper tuned model accuracy. The same kind of machine learning model can require different constraints, weights or learning rates to generalize different data patterns. These measures are called hyperparameters, and have to be tuned (hyper tuned) so that the model can optimally solve the machine learning problems. So hyper tuned model accuracy (hyper tuned r2 score) is enhanced model accuracy. Generally, MAE, MSE, RMSE and r2 score of all the models introduced improves with model hyper tuning.

V. CONCLUSION

In this paper, we proposed the rapid feature selection method based on feature ranking: the rankings of important variables obtained from feature engineering differ slightly, whereas the members of the low ranked features are almost the same. Since empirical rule is solved mathematically, the reason our method is successfully becomes clear. Ignoring zero important features and well correlated between themselves excluding the target features, helps if we have a lot of features reducing running time it takes and feature dimensionality. Then select variables with largest number of negative correlation. 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. Compared with the usual random forest algorithm and extremely random forest algorithm, the improved algorithm has higher accuracy when the amount of data is small, and the algorithm weakens the multi-valued bias problem, and does not need the logarithm operation.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 04, April-2021

ACKNOWLEDGMENT

I thank Leo Breiman the developer of Random forest algorithm for valuable insights into the inner workings of the method published at his web site and my research supervisor Professor Wu Zhi Feng including staffs of Tianjin University of Technology and Education, School of Information Technology and Engineering for giving me the opportunity to do research and providing invaluable guidance and support throughout this research.

REFERENCES

[1] 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
[2] A. Paul and D. Mukherjee, "Mitosis detection for invasive breast cancer grading in histopathological images," Image Processing, IEEE Transactions on, vol. 24, no. 11, pp. 4041–4054, Nov 2015
[3] WANG Zijing˙LIU Yu.An improved ID3 algorithm for decision tree[J]. Modern Electronics Technique,2018,41(15):39-42
[4] Breiman L.Bagging forests [J]. Machine Learning,2001, 45(1):5-32
[5] A. Criminisi and J. Shotton, Decision forests for computer vision and medical image analysis. Springer Science & Business Media, 2013
[6] 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
[7] 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.
[8] T. G. Dietterich, "Ensemble methods in machine learning," in Multiple classifier systems. Springer, 2000, pp. 1–15.
[9] Zijng Wang, Yu Liu, Lu Liu. A new way to choose splitting attribute i n ID3 algorithm[J]. 2017 IEEE 2nd Information Technology, Networkin g, Electronic and Automation Control Conference. IEEE, 2017:659-663
[10] Breiman L. Bagging predictors [J]. Machine Learning, 1996, 24 (2): 12 3–140
[11] CHEN Mincheng, YUAN Jingling, WANG Xiaoyan.Parallelization of R andom Forest Algorithm Based on Discretization and Selection of Weak -correlation Feature Subspaces[J].Computer Science,2016,43(06):55-58+9 0
[12] Yu Liu , Lu Liu , Yin Gao , Liu Yang, An Improved Random Forest Algorithm Based on Attribute Compatibility. 2019 IEEE 3rd Information Technology,Networking,Electronic and Automation Control Conference (ITNEC 2019),pp.2558-2561
[13] WANG Zijing,LIU Yu.A new attribute selection method of ID3 algorit hm for decision tree[J].Modern Electronics Technique,2018,41(23):9-12.
[14] Angshuman Paul, Dipti Prasad Mukherjee, Senior Member, IEEE, Prasun Das, Abhinandan Gangopadhyay, Appa Rao Chintha and Saurabh Kundu. Improved Random Forest for Classification.
[15] 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
[16] L. Breiman, "Random forests," Machine learning, vol. 45, no. 1, pp. 5–32, 2001.