URSI AP-RASC 2019, New Delhi, India; 09 - 15 March 2019

An Improved Random Forest Algorithm for classification in an imbalanced dataset.

Christy Jose* (1) and Gopakumar G(1)
(1) Department of Computer Science and Engineering, Amrita Vishwa Vidyapeetham, Amritapuri Campus

Abstract

Nowadays machine learning algorithms are being used extensively in industrial applications. Many a times these algorithms are modified and fine tuned so as to improve the current products and get better results. In this paper, we analyse an industrial problem that was put forward in the 'IDA 2016 challenge' and propose an improved solution over the best solution identified as part of the challenge.

1 Introduction

Industry based machine learning challenges or contests are popular these days. They present current problems in the industry and encourage researchers, students and enthusiasts to come up with innovative solutions.

This manuscript addresses the solution for an industrial challenge in autmotive industry: 'the IDA 2016 industrial challenge' [1]. The objective of this challenge is to create a prediction model for judging whether there is an imminent chance for the failure of a vehicle component. The dataset is provided by Scania and is derived from their range of heavy trucks. The component under study is the APS (Air Pressure System). There are two classes in the dataset, the positive class which corresponds to trucks with failure in the APS and the negative class which corresponds to trucks with failure in some other component. Altogether there are 76000 samples in the dataset, which includes 60000 training samples and 16000 testing samples.However, the distribution of the samples among the two classes is highly imbalanced with a large number of samples favouring the negative class and there is also an abundance of missing values.

Researchers world-wide taken up the IDA industrial challenge and had produced great results [2, 3, 4]. The paper by Camila F. Costa and Mario A. Nascimento [2] was selected as the winner of the challenge in which they proposed a Random Forest (RF) based solution to the problem and has shown that their solution has outperformed the solutions based on Support Vector Machines (SVM), Logistic Regression (LR) and K-Nearest Neighbours (K-NN). In this manuscript, we show that a carefully designed RF further improves the classification accuracy substantially both in terms of false positive rate (FPR) and false negative rate (FNR). We have achieved this result by carefully improving the strength and reducing the correlation of the RF.

This manuscript is organised such that section 2 provides necessary theory and mathematical background of RF. We introduce the measures for correlation and strength of trees in RF that determines its classification power. Also, we discusses how these measures are manipulated to produce better classification performance. Section 3 provides the implementation details including the procedure we followed for growing the trees. The superior performance of the proposed method is then provided in section 4 by comparing it with the state-of-the-art results in the literature.

2 Methodology

As we have mentioned in the introduction, the problem that we are addressing in this paper is basically a classification problem. There are many classification algorithms like SVM, LR etc which are originally designed to work with linearly separable data [5]. For non-linearly separable data, these algorithms are extended by introducing additional techniques like kernel methods for SVM [6]. On the other hand decision tree is very intuitive classifier that are widely used with non-linear separable data producing high classification accuracy. However they are prone to overfitting. A RF is an ensemble of decision trees and does not have the adverse effects of overfitting. This must be why RF preformed the best when compared to other algorithms as seen in the paper [2].Therefore we decided to start of with the RF algorithm and come up with modifications, in the hopes of getting a better result.

Before devising a strategy to improve the results produced by RF, the factors that govern its accuracy and the factors that helps to avoid overfitting in RF have to be studied. Following subsections discusses these factors, where we denote RF as a collection of tree-structured classifiers $h(x,\Theta_k)_{k=1}^{n}$,where the $\{\Theta_k\}$ are independent identically distributed random vectors. Here $h(x,\Theta)$ denotes a tree classifier where x is the input vector and Θ is a bootstrapped sample from the dataset.

2.1 Generalisation Power of RF

Overfitting occurs when we fail to develop a generic model based on a dataset. This increases the probability of getting errors when the model is applied on unseen data. Thus the extend of overfitting can be measured by calculating the

generalization error. We are going to show that the generalisation error in RF has a limiting value, there by implying that RF has less chance of overfitting the data.

The generalization error for a collection of classifiers is defined in terms of a margin function. Let $\{h_k(x)\}_{k=1}^{n}$ be a collection of classifiers. And, let Y, X be a random vector sampled (from some distribution) from the training data. Here Y is the set of class label corresponding to the input vector in X. We now define the margin function of a collection of classifiers as

$$mg(X,Y) = \left[\frac{\sum_{l=1}^{k} I(h_l(X)=Y)}{k}\right] - \max_{j\neq Y}\left[\frac{\sum_{l=1}^{k} I(h_l(X)=j)}{k}\right] \quad (1)$$

where I(.) is the indicator function.

As stated above in equation 1, the margin functions gives us an idea of by how much the average votes for the correct class differs from the average votes for any other class. If $mg(X,Y) > 0$, the collection of classifiers votes for the correct class. If $mg(X.Y) < 0$, the collection of classifiers votes for an incorrect class. Also higher the margin, the more the confidence in the classification.

Now the generalization error can be defined as

$$PE = P_{X,Y}(mg(X,Y) < 0) \quad (2)$$

which is the probability that the value of the margin function is less than zero. In the context of random forests, each classifier $h(x)$ is $h(x,\Theta)$. Thus the margin function for a Ramdom forest can be defined as

$$mr(X,Y) = P_\Theta(h(X,\Theta)=Y) - \max_{j\neq Y}[P_\Theta(h(X,\Theta)=j)] \quad (3)$$

As the number of trees increases, PE converges to

$$P_{X,Y}(P_\Theta(h(X,\Theta)=Y) - \max_{j\neq Y}[P_\Theta(h(X,\Theta)=j)] < 0) \quad (4)$$

This relation follows from the Strong Law of Large Numbers and its proof is given in the paper [7]. This means that the generalization error has a limiting value and that random forests do not over-fit the data.

2.2 Factors affecting Accuracy

The accuracy of RF increases as the generalization error decreases. To study how accuracy relates to the individual trees in RF, we redefine generalization error in terms of strength and correlation between the trees. We need the decision trees in RF, to have high confidence in classification and at the same time to look at the data in different way when compared to other trees in RF. Thus we need to have trees with maximum strength and minimum correlation in a RF.

There is an established result that says the generalisation capabilty of RF can be increased by improving strength of individual trees and reducing correlation between trees in RF[7]. It is defined as

$$PE \leq \bar{p}\frac{(1-s^2)}{s^2} \quad (5)$$

where PE denotes the generalization error, $\bar{p}$ denotes the mean correleation and s denotes the strength of RF.

Note that the conclusions we made about the generalization error and its relationship between strength and correlation is based on the assumption that the RF contains a large number of trees. However there is a research work that experimently proves that these properties also hold for smallers RFs with number of trees between 50 to 200 [8].

Now we look into how strength and correleation are formulated and how we can manipulate them to suit our purpose.

2.2.1 Improving the Strength

As we inferred eariler from equation 5, one of the ways to improve the accuracy of the RF is to increase its strength. The strength can be thought of as a measure of how accurate the individual trees are in predicting the correct class. It is be defined as

$$s = \mathbb{E}_{X,Y}(mr(X,Y)) \quad (6)$$

Here $\mathbb{E}$ denotes expectation operator.

From equation 6 it is clear that to increase strength, we must increase the value of the margin function. This can be achieved by increasing the accuracy of the individual trees.

The accuracy of the tree depends on the decisions made by the tree at each of its nodes. This is computed using impurity measures of the node.

We use Gini's Diversity Index as the measure of impurity of each node in the tree. It is defined as

$$I_t = 1 - \sum_i (P(i))^2 \quad (7)$$

where I_t denotes the impurity measure of node t, i denotes a class at node t and $P(i)$ denotes the relative measure of occurance of class i at node t. The sum is calculated over all classes i that reach node t. If the node has only one class, its impurity will be zero and is called a pure node. Otherwise, the impurity will be a positive value.

Now to decide what decision is to be made at a node we calculate the impurity gain of all possible decisions that can be made at a node. The decision that gives the best impurity gain is assigned to that node.

The impurity gain at a node t for a pirticular decision is given as

$$\Delta I_t = P(T - T_u) \times I_t - P(T_l) \times I_l - P(T_r) \times I_r \quad (8)$$

where ΔI_t is the impurity gain at node t, T is the set of all observation indices at node t. T_u is the set of indicies in T which have missing values. l and r are left and right child nodes resulting from the split at node t by the decision made. T_l and T_r are the set of observations at the nodes l and r respectively.

Here $P(T)$ denotes the sum of the probablilites of all observation indicies in the set T. It is defined as

$$P(T) = \sum_{j \in T} w_j \quad (9)$$

where w_j denotes the weight of observation j in T. w_j is equal to $1/n$ (unless specified otherwise), where n is the number of observations in T.

2.2.2 Reducing the Correlation

As inferred from equation 5, another way to improve the accuacy of the RF is to reduce the correlation. The correlation can be thought of as the measure of similarity between two trees in RF.

The correlation is defined as

$$p(\Theta, \acute{\Theta}) = \frac{\text{cov}_{X,Y}(rmg(\Theta, X, Y), rmg(\acute{\Theta}, X, Y))}{sd(\Theta) sd(\acute{\Theta})} \quad (10)$$

where $p(\Theta, \acute{\Theta})$ denotes the correlation, $rmg(\Theta, X, Y)$ denotes the raw margin function and $sd(.)$ denotes the standard deviation. The raw margin function, is defined as

$$rmg(\Theta, X, Y) = I(h(X, \Theta) = Y) - I(h(X, \Theta) = \hat{j}(X, Y)). \quad (11)$$

where $I(.)$ is the indicator function and $\hat{j}(X, Y)$ denotes the most probable predicted class other that Y (the true class), which is calculated as.

$$\hat{j}(X, Y) = \arg\max_{j \neq Y} (P_\Theta(h(X, \Theta) = j) \quad (12)$$

Rewriting the margin function for RF in terms of $\hat{j}(X, Y)$, we get

$$\begin{aligned} mr(X, Y) &= P_\Theta(h(X, \Theta) = Y) - P_\Theta(h(X, \Theta) = \hat{j}(X, Y)) \\ &= \mathbb{E}_\Theta(I(h(X, \Theta) = Y) - I(h(X, \Theta) = \hat{j}(X, Y))) \\ &= \mathbb{E}_\Theta(rmg(\Theta, X, Y)) \end{aligned} \quad (13)$$

This means that the margin function is the expected value of the raw margin function we defined in equation 11.

Thus from equation 10, we can see that to reduce the correlation we have to reduce the variance between the trees in RF. Bootstrap Aggregating (or Bagging) [9] is a good technique which can applied to reduce the variance without causing much changes in the bais. In this process each tree is grown using a subset of the dataset. The subset is generated by randomly choosing samples (with replacement) from the dataset. It has also been expreimently shown that such types of ensemble methods can improve the accuracy of classifiers [10].

Further improvements in accuracy can be achieved by growing a seperate RF using misclassified data and combining it with the original RF. The intuituion behind this is the secondary RF will capture certain features missed out by the original RF and thus improve the accuracy.

Table 1. Classification Results on Scaina APS Training Dataset

Classifier	% Miss-classification Rate
Random	FPR: 50% FNR: 50%
SVM	FPR: 1.15% FNR: 13.5%
LR	FPR: 2.36% FNR: 9.5%
K-NN	FPR: 2.94% FNR: 6.5%
RF	FPR: 3.74% FNR: 3.7%
RF (our)	FPR: 2.8% FNR: 2.7%

Source: The first five rows were sourced from paper [2]

2.3 Addressing Missing Values

The dataset provided as part of the challenge contains a lot of missing values. Ignoring the samples with missing values will leave us with very few training samples. The test samlples also contains missing values and so it is necessary that we find ways to impute the missing values. Also it is necessary that we predict the missing values properly, otherwise the RF algrithm may pick up the wrong pattens from the data and perform poorly. K-Nearest Neighbours(KNN) is a very good algorithm for prediciting data values. By defining the value of K properly we can get a good estimate of the actual values.

3 Implementation

The RF is created by growing N number of trees. Each individual tree is grown using boostrap-aggregrated [9] data. Also a seperate RF with S trees is grown using the misclassified data. The final result is computed by combining the weighted results of the two RFs. A sample is calsified as belonging to the negative class only if the combined confidence is more than 0.95. This is done to neutralise the effects of the high imbalance in the dataset.

For imputing missing values we have used KNN algorithm with K = 33.

Table 2. Classification Results on Scaina APS Test Dataset

Classifier	FPs	FNs	Cost
First	542	9	9920
Second	490	12	10900
Third	398	15	11480
Our	420	14	11200

Source: The first there rows were sourced from IDA 2016 website [1]

Table 3. Preformance Measures

Classifier	Precision	F-measure	MCC
First	0.40	0.57	0.61
Our	0.46	0.62	0.65

Source: Computed with the avaliable data.

4 Result

We have used RF with 50 trees as the primary classifier and RF with 25 trees as the secondary RF. The RFs are constructed from the training samples of the dataset.To get an estimate how well the RFs would perform on the test samples we have used 10 fold cross-validation. The average False Positive Rate (FPR) and False Negative Rate (FNR) obtained during cross-validation are shown in Table 1. Our classifier also performs at par the best classifier in the test dataset with an avaerage of 11 False Negatives (FN) and 420 False Positives (FP).The results obtained by the top 3 classifiers in the IDA Industrial Challenge is compared with the results we obtained in Table 2. Here the cost for FPs is taken as 10 and FNs is taken as 500. The precision, F-measure and Matthews correlation coefficient (MCC) of the classifier is also measured and compared with the top classifier in the challenge, as shown in Table 3. As we can see our classifier is beter in all the three above mentioned measures. Even still these are just intermediate results and we are currently working on improving the classifier to get better results, more specifically an improvement in FNs count.

References

[1] "Industrial challenge." [Online]. Available: https://ida2016.blogs.dsv.su.se

[2] C. F. Costa and M. A. Nascimento, "IDA 2016 industrial challenge: Using machine learning for predicting failures," in *Advances in Intelligent Data Analysis XV - 15th International Symposium, IDA 2016, Stockholm, Sweden, October 13-15, 2016, Proceedings*, 2016, pp. 381–386. [Online]. Available: https://doi.org/10.1007/978-3-319-46349-0_33

[3] V. Cerqueira, F. Pinto, C. R. de Sá, and C. Soares, "Combining boosted trees with metafeature engineering for predictive maintenance," in *Advances in Intelligent Data Analysis XV - 15th International Symposium, IDA 2016, Stockholm, Sweden, October 13-15, 2016, Proceedings*, 2016, pp. 393–397. [Online]. Available: https://doi.org/10.1007/978-3-319-46349-0_35

[4] E. C. Ozan, E. Riabchenko, S. Kiranyaz, and M. Gabbouj, "An optimized k-nn approach for classification on imbalanced datasets with missing data," in *Advances in Intelligent Data Analysis XV - 15th International Symposium, IDA 2016, Stockholm, Sweden, October 13-15, 2016, Proceedings*, 2016, pp. 387–392. [Online]. Available: https://doi.org/10.1007/978-3-319-46349-0_34

[5] S. Jaysri, J. Priyadharshini, S. Palaniappan, and P. Kumar, "Analysis and performance of collaborative filtering and classification algorithms," vol. 10, pp. 24 529–24 540, 01 2015.

[6] K. Soman, R. LOGANATHAN, and V. AJAY, *Machine Learning with SVM and Other Kernel Methods*. PHI Learning, 2009. [Online]. Available: https://books.google.co.in/books?id=-QFhFUNxIZwC

[7] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct 2001. [Online]. Available: https://doi.org/10.1023/A:1010933404324

[8] S. Bernard, L. Heutte, and S. Adam, "A study of strength and correlation in random forests," in *Advanced Intelligent Computing Theories and Applications*, D.-S. Huang, M. McGinnity, L. Heutte, and X.-P. Zhang, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 186–191.

[9] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, Aug 1996. [Online]. Available: https://doi.org/10.1023/A:1018054314350

[10] K. Lakshmi Devi, S. Palaniappan, and P. Kumar, "Tweet sentiment classification using an ensemble of machine learning supervised classifiers employing statistical feature selection methods," pp. 1–13, 01 2015.