KNOWLEDGE DISTILLATION FROM RANDOM FOREST

Clide Dcosta

***Abstract*—Over the years there has been a number of significant works that combine two more classifiers to produce a single classifier. The resulting classifier, referred to as an ensemble classifier, is a set of classifiers whose individual decisions are combined into one predictive model in order to decrease variance, bias or improve prediction. The accuracy predicted from ensemble classifiers are more accurate than of the individual classifiers that make them up. This paper presents the accuracy obtained by random forest, one of the techniques of generating an ensemble of classifiers. The random forest classifier is trained against wine quality,glass classification and heart disease data set. The disadvantage of an ensemble of classifiers is that predictions produced by them are complex and may be too computationally expensive to allow deployment to a large number of users.(Geoffrey Hinton, Oriol Vinyals, Jeff Dean 2015). It is possible to compress the knowledge in an ensemble into a single model which is easier to deploy.**

I. INTRODUCTION

Recent advancements in machine learning algorithms have been nothing short of phenomenal. Better models are being built that give better accuracy and are better performing as the years go by. This has been possible due to a number of theoretic and algorithmic breakthroughs in the last decade or so , facilitated by significant technological advancements in computer hardware and software (George Papamakarios 2015). Large Ensembles which are a combination of many different individual classifiers are an example of a particular trend that has been introduced in recent years. One famous ensemble method is Random forest classifier.

Random forest is an algorithm for classification that uses an ensemble of classification trees. Each of the classification trees is built using a bootstrap sample of the data. At each split the candidate set of variables is a random set of variables. The random forest classifier uses bagging(bootstrap aggregation) and Random variable selection for tree building. Each tree is grown fully so as to obtain trees with low bias and low variance. The bagging and random feature selection techniques are used to produce trees in which the correlation is minimum.

The advantage of Random forest classifiers over other classifiers is that it runs efficiently on large databases and it produces the most accurate predictions when compared to other classifiers. One of the most important features of Random forest is that it gives an estimate to as which of the variables are the most important for classification. This is called Variable importance. It is the most efficient when it comes to handling missing data and maintains a high degree of accuracy when a large proportion of data is missing.

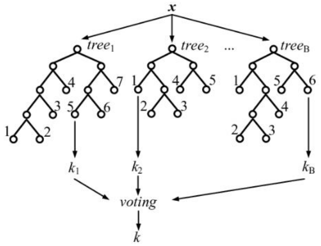

Fig. 1. Random forest architecture

Random forest being an ensemble method results in an increase in model size and complexity. As a result of its large size and complexity, a number of significant practical challenges arise (George Papamakarios 2015). Random forest being a collection of decision trees have several hidden layers which results in evaluation time of model to increase especially if the n_estimators (number of decision trees) are large. Due to its large size and sophisticated structure, Interpretability will reduce and integration into larger systems will be hard. Size and complexity may present models with power, but they can also make them cumbersome and difficult to use (George Papamakarios 2015).

Suppose we have a model that give predictions with a high degree of accuracy, but is cumbersome and inconvenient to use. Imagine we can build another model from the cumbersome model that meets performance criteria in terms of accuracy and efficiency but is no longer cumbersome and is found to be convenient. In this paper, we will put in theory this fact to construct convenient model from a cumbersome one. This procedure is called knowledge distillation. Here, a small model is trained to mimic a pre-trained larger model(ensemble of models).

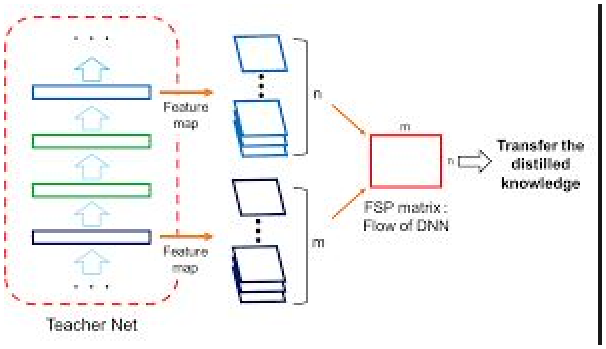

Fig. 2. Knowledge distillation architecture

Clide Dcosta (e-mail: cd18224dcosta@gmail.com)

II. BACKGROUND

In ensemble classification multiple classifiers are used to predict the model and the accuracy generated is significantly much higher that of the individual classifiers. A voting technique is used to determine the class label for the variable being considered. A simple but effective voting scheme is majority voting(Lam &Suen,1997). In majority voting each classifier in the ensemble is asked to predict the class of the unlabeled instance. Once all the votes have been tallied up, the classifiers having the maximum number of votes is returned as the predicted class of that unlabeled instance. Another alternative voting scheme used is called Veto voting. Here one single classifiers vetoes the decision of other classifiers(Shazad & Lawson,2012,Sun and Dance 2012). The most common voting scheme that recently burst into existence was trust based veto voting (Shahzad and Lavesson 2013) which is an extension of veto voting. The trust of each classifier in the ensemble is determined and used to find out whether a classifier or set of classifiers can veto the decision.

The most common ensemble methods used are Bagging, Boosting and Stacking. Boosting is defined as building a sequence of classifiers in an incremental manner where each classifier works on the instances which are incorrectly classified in the previous cycle. Stacking is an ensemble model ,where a new model is trained by the combined predictions of two or more previous models. The predictions from the model are used as input for each sequential layer and combined to form a new set of predictions. Bootstrap aggregation is most simple and powerful ensemble method. This method is used to reduce the variance for algorithm with high variance. One such algorithm which has high variance is decision trees. Random forest is the main representative of bagging (Breiman 2001)

Breiman [2001] was the first to provide a general definition of a Random forest and also proposed what is arguably the most popular Random forest algorithm. Concretely, Breiman defines an RF as follows:

> A random forest is a classifier consisting of a collection of tree-structured classifiers {t(x, Θb), b = 1, ..., B} where the {Θb} are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input x.

Breiman (2000) proposed two different ways for creating an ensemble of base classifier for a classification problem. One way of creating an ensemble method is by using technique such as Adaboost(Freund and Schapire,1996) and Margin Classifiers(Mason 2000). These methods create ensembles by reweighing the training data set iteratively without having any randomness involved in the process. A weighted voting is then used to assign the test data set to its class. Another way that was suggested by Breiman(2000) to create an ensemble of classifiers from individual classifiers was through Bagging(Breiman 1996), Random split decision(Diettrich,1998) and Random feature selection(Amit and Geman 1997,Breiman 1999). These techniques employ randomness on either of the training data or the features in the data set and uses unweighted voting to assign the classes to a new feature.

Knowledge distillation was introduced only recently by Hinton et al.(2015), who used it to mean model compression. Knowledge distillation is effective to train the small and generalisable network models for meeting the low-memory and fast running requirements. Knowledge distillation compacts deep networks by letting small student network learn from a large teacher network(Silvia L.Pintea 2016). In this paper we look at how models are built into random forest and transfer the knowledge of a cumbersome model into building a convenient model.

III. METHODOLOGY

In this paper we will be looking at three distinct data sets. Our initial phase of the project includes pre-processing the data sets . This includes loading the data sets and the necessary libraries. During this stage we will check the data sets for any missing values. If present we use a suitable method like mean or mode to replace these values. We will also build correlation matrices to give a depiction to as how each of the predictor variables depend on the target variables. Outliers are detected in the data set if present . The most important task when it comes to pre-processing is to find out the relevant attributes that are needed to improve the prediction of our model . In this paper we use two different ways to select these variables, feature selection and principal component analysis.

The second phase of our project involves running random forest classifier on the dataset. The probabilities of each class present in the dataset are retrieved using this classifier. Using a binning method we convert the binary classification of our dataset into a multi classification that includes the probabilities along with the class of the attributes. We then train a new small distillation model to make the same predictions as this ensemble. We then train the distillation model on softened target probabilities by raising the temperature used in softmax calculations. The reason behind this being that, it allows us to transfer more information per training examples from the cumbersome to the distillation model. Afters we train a decision tree classifier on the new data set and compare the accuracy on the original dataset. We also compare the performance of the new data set with other machine learning algorithms like SVM, K-N nearest neighbors, Naive Bayes etc. Let us look at few terminologies mentioned here.

A. *Random forest*

Random forests is an ensemble classifier that consists of many decision trees as input and outputs the class that is the node of the class's output by individual trees. The method combines Breiman's bagging idea and the random selection of features. In Random forest we analyze the data by first selecting a target attribute. In applications of business this target value may be the final status of a loan or credit card account, for detection of fraud status of insurance claim. In a large data set with numerous features it becomes essential to reduce the number of attributes so that only the relevant variables are used to build and predict the model. This is called feature selection. The key advantage of this random forest variable importance is that they cover the impact of each predictor variable individually as well as in multivariate interactions with other predictor variables.

Once the target variables have been identified, Random forests begin by growing a CART-like decision tree. Here a bootstrap sample is used instead of the traditional training data. The bootstrap sample includes only 2/3 of training data. The selection of the variable to split the node in the construction of the tree is chosen at random. Finally, the decision tree is grown out to the largest possible size and left unpruned.

B. *DECISION TREES*

Decision trees are individual learners that are combined. They are one of the most popular learning methods commonly used for data exploration. One type of decision tree is called CART. . . classification and regression tree. CART . . . greedy, top-down binary, recursive partitioning, that divides feature space into sets of disjoint rectangular regions.

Assume that we have an input dataset X with n features (feature 1,2,...,n), and each instance of the dataset has a classification (say, class A, B, and C). A decision tree classifier trained on this dataset with depth = 2 will *approximately* look like this:

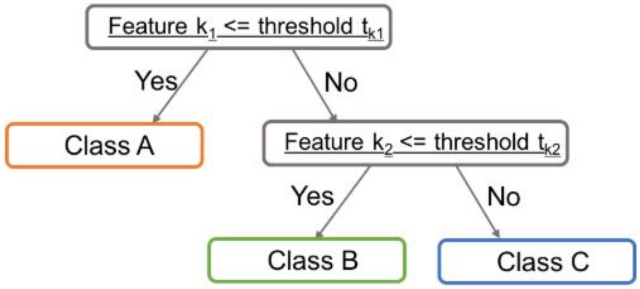

Fig. 3. Decision tree classificattion

To make a prediction, the decision tree first compares an instance's feature $k_1$with threshold t_{k1}. If $k_1 \leq t_k \leq t_{k1}$, then the instance is classified as "Class A". If $k_1 > t_{k1} > t_{k1}$, then the decision tree checks if this instance's feature $k_2$is less than or equal to threshold t_{k2} If yes, then the instance is classified as "Class B", otherwise it is classified as "Class C". The node on the very top is called *root node* (depth = 0), and the nodes that do not have any children are called *leaf node*.

1) GINI IMPURITY: There is another node attribute in the above figure: Gini impurity. Gini impurity describes how "pure" the sample composition is in a node. For example, if a node has 0 samples in class_0 and class_1, but 20 samples in class_2, Gini impurity will be 0. The mathematical definition of gini impurity is:

$G = 1 - \sum_{K=1}^{n} p_k^2$

where $p_k$is the ratio of class k instances in the node. For the bottom left node in the above figure, we have

$G = 1 - (0/46)^2 - (6/46)^2 - (40/46)^2 \approx 0.227.$

Gini impurity serves as a criterion for splitting the nodes in decision trees.

2) TRAINING ALGORITHM: The Classification and Regression Trees (CART) algorithm works by splitting a node into two children nodes at a time. The splitting criterion consists of a feature $k$and a threshold for this feature t_k, which are selected by minimizing an impurity cost function:

$J(k,t_k) = G_{left} \times m_{left}/m + G_{right} \times m_{right}/m$

This is a weighted sum of the gini impurity of the left and right child usually stops if the tree has reached its user-defined maximum depth, or if the after-split impurity $G_{left} \times m_{left}/m + G_{right} \times m_{right}/m$ is bigger than the parent node's impurity G_{parent} but other stopping criteria exist, too.

C. *KNOWLEDGE DISTILATION*

Knowledge distillation is model compression method in which a small model is trained to mimic a pre-trained ,larger model. This training setting is sometimes referred to as "teacher-student", where the large model is the teacher and the small model is the student.

In distillation, knowledge is transferred from the teacher model to the student by minimizing a loss function in which the target is the distribution of class probabilities predicted by the teacher model. However, in many cases, this probability distribution has the correct class at a very high probability, with all other class probabilities very close to 0. As such, it doesn't provide much information beyond the ground truth labels already provided in the dataset. To tackle this issue, Hinton et al., 2015 introduced the concept of "softmax temperature". The probability p_i of class i is calculated from the logits as:

$p_{i=}exp(z_i/t)/\sum_j exp(z_j/t)$

where T is the temperature parameter. When T=1 we get the standard softmax function. As T grows, the probability distribution generated by the softmax function becomes softer, providing more information as to which classes the teacher found more similar to the predicted class. When computing the loss function vs. the teacher's soft targets, we use the same value of TT to compute the softmax on the student's logits. We call this loss the "distillation loss".

The overall loss function, incorporating both distillation and student losses, is calculated as:

$L(x;W) = \alpha * H(y,\sigma(z_s;T = 1)) + \beta * H(\sigma(z_t;T = \tau),\sigma(z_s,T = \tau))$

where x is the input, W are the student model parameters, y is the ground truth label, H is the cross-entropy loss function, σ is the softmax function parameterized by the temperature T, and α and β are coefficients. z_s And z_t are the logits of the student and teacher respectively.

IV. DATA SET DESCRIPTION

In this paper we will inspect three distinct data set and try to instill the knowledge of distillation in it. The data sets in question can be found on the UCI repository. These data sets are a representation of binary classification. The three data sets chosen are red wine quality, glass classification and heart disease.

The red wine quality data set has a total of 13 variables i.e. 12 predictor variables and one target variable which is quality.

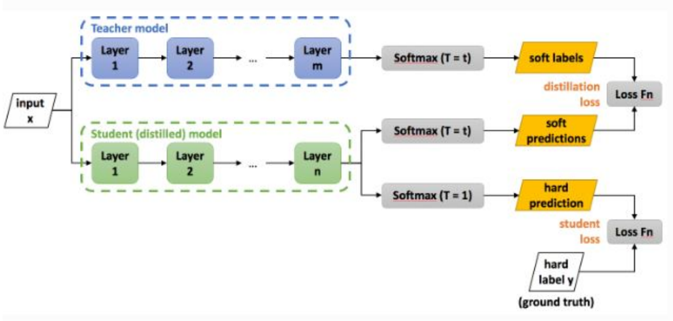

Fig. 4. Knowledge distillation architecture

The target variable has a range of values ranging from 0 to 10 with zero being the wine with the worse quality and 10 being the one with the highest quality. Some of the predictor variables include citric acid , volatile acid , fixed acidity etc. Through the process of feature selection we can find out the most relevant attributes needed to build our model.

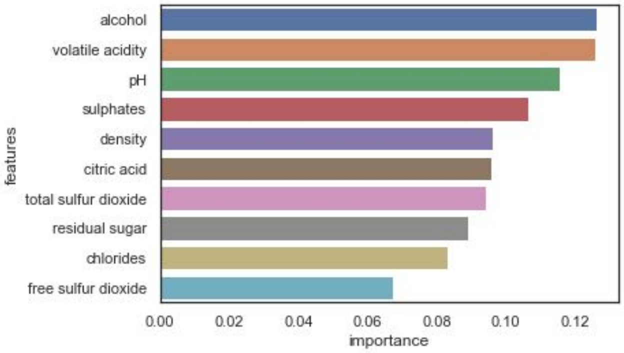

Fig. 5. Feature selection using variable importance for wine quality data set

The heart disease data set has 12 predictor variables and one target variable which has values 1 and 0 . The target variable depicts whether the person has heart disease (1) or not(2). Using feature selection the relevant attributes are found

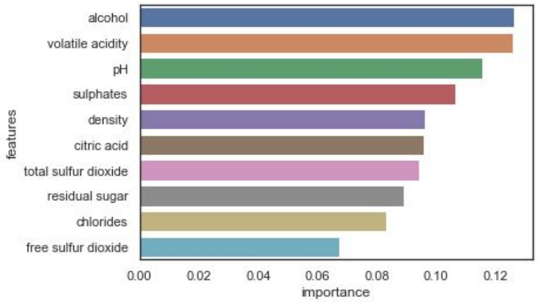

Fig. 6. Variable selection for heart disease data set

The glass data set contains 10 attributes including id. The response is glass type(discrete 7 values).Using feature selection we can retrieve the relevant attributes.

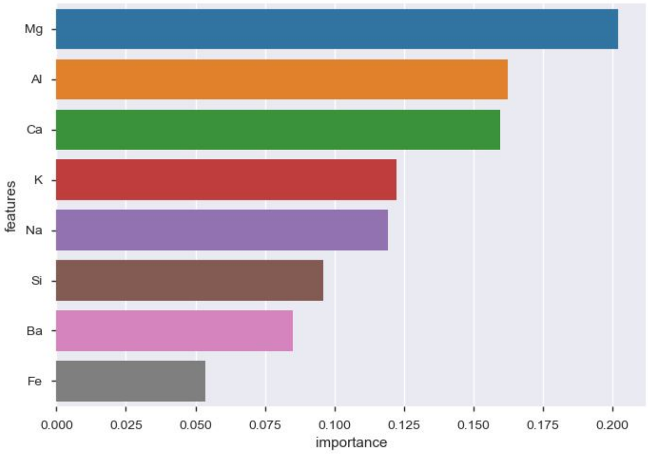

Fig. 7. Variable selection for glass classification dataset

V. Experiments

The initial part of the project involves all the important aspects of pre-processing . We load the necessary libraries and the data set, check for correlation among the variables, check for missing values in our dataset if there are any and replace them, check for outliers among the data . Once the initial process of data cleaning is over our next important step for pre-processing involves selecting the best features to best fit our model to achieve a high degree of accuracy. Elimination of unneeded variables that lower the efficiency is required . We achieve this through either through Principal Component Analysis (PCA) or Feature selection.

A. *PRINCIPAL COMPONENT ANALYSIS*

Principal component analysis is a statistical procedure to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables. Each of the principal components are chosen in such a way so that it would describe most of the still available variance and all these principal components are orthogonal to each other. In PCA the first principal component has maximum variance, while the second principal component has the second maximum and so on. PCA can be used to find inter-relation between variables in the data. As the number of variables decreases PCA makes further analysis simpler. The main objective of PCA is dimension reduction and to select a subset of variables from a larger set ,based on which the original variables have the highest correlation with the principal amount.

B. *FEATURE SELECTION*

Feature selection also known as variable selection is the automatic selection of attributes in data that are most relevant to the predictive modeling problem . Feature selection methods are used to identify and remove unneeded, irrelevant and redundant attributes from the data that do not contribute to the accuracy of a predictive model or may in fact decrease the accuracy of the model. The objective of variable selection is three-fold:Improving the prediction performance of the predictors, providing faster and more cost effective predictors, and providing a better understanding of the underlying process that generated the data

Feature selection is different from dimensionality reduction. Both methods seek to reduce the number of attributes in the dataset, but a dimensionality reduction method do so by creating new combinations of attributes, whereas feature selection methods include and exclude attributes present in the data without changing them.

VI. Discussion

The three data sets are first put though the data pre-processing stage where the data set and necessary libraries are loaded . Data cleaning and data visualization techniques are used to get a better understanding of the data in use . Some of these techniques include checking for null values present and replacing them with an appropriate method like mean , mode if needed, building a correlation matrix to visualize the dependencies of the predictor variables against the response variable. Checking for any outliers in the data is also an essential step in the pre-processing phase. Statistical analysis on the data is performed . Finally, the data is split into training and test data sets.is split. Once the initial clean up task is completed, our next job is to run the random forest classifier against the data and find the accuracy of the predicted model and to remove the probabilities of each class present in it. Once the probabilities have been acquired , we instill the concept of knowledge distillation on the new dataset so as to reduce its size and complexity . A decision tree classifier trained on the new data set consisting of these generated probabilities. Other machine learning algorithms are implemented on this data set and their accuracy are compared.

VII. Conclusion

In this initial part of the project we load the data sets and perform the preprocessing tasks . We separate out the relevant attributes needed to improve the performance of our model. Outliers are detected and dealt with. Missing values are also dealt . This completes the initial phase of our project . In the final implementation we will have a better understanding as how the concept of knowledge distillation is used.

VIII. References

Geoffrey Hinton, Oriol Vinyals,Jeff Dean ,Distilling the knowledge in a Neural Network (2015)

George Papamakarios,Distilling Model Knowledge(2015)

L. Breiman, *bagging predictors Machine Learning*, vol. 26, pp. 123-140, 1996.

E. Bauer, R. Kohavi, "An empirical comparison of voting classification algorithms", *Machine learning*, vol. 36, pp. 105-139.

M.Pal Random forests for land cover classification(2005)

M. Pal, P. M. Mather, "Decision tree classifiers and land use classification", *Proceedings of the 27th Annual Conference of the Remote Sensing Society*, 12-14 September, London, UK, 2001.

Y. Freund, R Schapire, "Experiments with a new boosting algorithms Machine Learning", *Proceedings of the Thirteenth International conference*, pp. 148-156, 1996.

Juergen Gall, Nima Razavi, and Luc Van Gool,An Introduction to Random Forests for Multi-class Object Detection(2009)

Cristian Bucilˇ,Rich Karuna,Alexandru Niculescu-Mizil Model Compression(2015)

L. Breiman, Random Forests, 2000.

Amit Y., & Geman, D. (1997). Shape quantization and recognition with randomized trees. Neural Computation, 9(7), 1545–1588.

Shahzad, R. K., & Lavesson, N. (2012). Veto-based malware detection. In 2012 seventh international conference on availability, reliability and security (ARES), Prague, Czech Republic (pp. 47–54). New York City, NY: IEEE.

Sun,Y.-A.,&Dance,C.(2012).When majority voting fails: Comparing quality assurance methods for noisy human computation environment. CoRR. Retrieved from arXiv:1204.3516.

IX. Plan

TABLE I

Procedure	Time(Working day)	Finish By
Make project plan	3	14th feb 2019
Collect data and pre-process	2	18th feb 2019
Make intial project proposal	2	21st feb 2019
Build the training procedure and run Random Forest classifiers	2	3rd March 2019
Testing the perfomance and applying knowledge Distillation concept	3	10th March 2019
Problem modification	2	15th March 2019
Generate the final project in IEEE forat	7	1st April 2019