Implementation and Analysis of Random Forests

Jae Lee
School of Computing Science
Simon Fraser University
Burnaby BC V5A 1S6

Richard Mar
School of Computing Science
Simon Fraser University
Burnaby BC V5A 1S6

Robin White
School of Mechatronic Systems Engineering
Simon Fraser University
Surrey BC V3T 0A3

Abstract

TODO: Decide if we need this section.

1 Introduction

In machine learning, there is often a tug-of-war between bias and variance; having high accuracy to observed data but not to lose generalization (or over-fit) to unseen data. This is often referred to as the "bias-variance tradeoff" and it's consideration is a significant part of properly engineering machine learning algorithms. Often, regularization is used where there is an addition to the loss function to represent a cost to complexity, or in the way of neural networks, random dropout of neurons to force generalization [1]. Another method is to also use validation datasets so as to understand the loss from unseen data without actually testing on the test set. All of these methods add to the complexity of the algorithm and can also often lead to loss in accuracy of the training data.

In the early 90's, Tin Kam Ho from Bell Labs published a series of papers where he showed surprisingly that by combining independent learners in a unique way increased the accuracy of classifying handwritten digits monotonically; without suffering from over-adaptation to the training data. [2, 3, 4] The application of this method to decision trees in his '93 paper marked the introduction of Random Forests to the community. [2] Decision trees are simple yet effective classifiers, with high execution speed and easily relatable, however they are limited by their complexity for possible loss of generalization to unseen data. Some methods such as pruning have been used previously to try and increase generalization, however methods such as these usually come with a loss in accuracy toward training data. By using principles of stochastic modeling, Ho showed that tree-based classifiers could be arbitrarily expanded for increases in accuracy on unseen testing data without loss in training data accuracy. A characteristic which is still unique among machine learning classifiers. The concept is that multiple learners can compensate for the bias of a single learner and so trees are constructed from randomly selecting subspaces of the feature space. In this way, each tree generalizes in a different way.

Random Forests have been applied to a variety of machine learning tasks including classifications in ecology and geosciences, image segmentation in medical applications, business analytics, sporting analytics, as well as the unmentioned number of general data science applications. [5, 6, 7, 8, 9] In this report we aim to investigate the application of Random forests as classifier and regressor to a hockey player dataset. We present the general algorithm coded in python, as well as investigate the various parameters such as number of trees, number of features per tree, and splitting criteria comparing entropy and Gini impurity for classification, and variance for regression. We also compare results of our algorithm to that of sci-kit learn and Weka, and discuss key similarities and differences. We conclude with discussion on limitations and challenges in present work on Random forests.

1.1 Decision Trees

TODO: Outline general Decision tree algorithm.

1.2 Random Forests

TODO: Outline general Random forest algorithm. Pseudo code

1.3 Ensemble Learning

TODO: Decide what content to go where

Bootstrap Aggregating (Bagging)

Bootstrap aggregating or bagging is an ensemble learning technique that attempts to reduce variance of the model without increasing the bias by attempting to remove correlation between individual trees. Each tree is limited to evaluating only a random fractional sample of the actual dataset such that no two samples are the same. Bagging has been demonstrated empirically to improve model accuracy. [10]

From the actual dataset D of size n, k bootstrap samples,

$$D_1, D_2, ..D_k$$

are randomly selected with repetition from D.

Then the variance of the ensemble is the average of the sum of the individual trees' variances.

$$var(L) = \frac{\sum_{i=1}^{k} var(L_i)}{k}$$

where L is the ensemble of individual learners (trees).

Each datapoint in D has a probability of

$$1 - \left(\frac{1}{n}\right)^n = e^{-1} \approx 36.8\%$$

of not being selected in a sample D_i and therefore approximately 63.2% probability of being in a training set D_i.

2 Approach

TODO: Figure out what is supposed to be here.

2.1 Forest Size

TODO: Add content.

2.2 Tree Depth

TODO: Add content.

2.3 Splitting Criteria

Random Subspace Method

Random forest uses a modified splitting algorithm that attempts to further reduce correlation between individual trees. For example, if few attributes are strong predictors of the target label, these attributes will be selected in many trees leading to high correlation and greater generalization error. Generalization error of an ensemble converges to the following expression:

$$Generalization\ error \leq \frac{corr(1-s^2)}{s^2}$$

where corr is the average correlation among the trees and s is the average performance of individual classifiers. Thus, reducing correlation among the individual trees will also lower the generalization error.

Work by Ho has demonstrated that average tree agreement between trees is significantly lowered using the Random subspace method. [4] Estimating tree agreement between trees i and j as $s_{i,j}$

$$s_{i,j} = \frac{1}{n}\sum_{k=1}^{n} f(x_k)$$

$$where\ f(x_k) = \begin{cases} 1 & if\ class\ decision_i(x_k) = class\ decision_j(x_k) \\ 0 & otherwise \end{cases}$$

Ho's result showed average of $s_{i,j}$ of random subspaces method was lower than that of bootstrapping and boosting methods alone. [4]

Thus, limiting a tree's evaluation to only a fixed size subset of the actual features and randomizing the elements in the subset during each splitting process helps to reduce correlation among each trees. The modified splitting algorithm will then split on a single feature with the best information gain ratio or Gini impurity to reduce correlation among trees.

2.3.1 Entropy

TODO: Add content.

2.3.2 Gini Index

Gini impurity index measures the probability of incorrectly labeling an element if it was randomly labeled according to the distribution of labels in the leaf node. Gini impurity and Entropy are analogous. Gini impurity is less computationally expensive since it doesn't require calculating logarithmic functions.

Given k classes and fraction of elements in the leaf node with class i, p_i, the Gini impurity can be calculated as:

$$G(p) = \sum_{i=1}^{k} p_i \sum_{j=1}^{k} p_j = \sum_{i=1}^{k}(p_i - p_i^2) = 1 - \sum_{i=1}^{k} p_i^2$$

2.4 Custom Improvement?

3 Experiments

TODO: Add content.

The ability of Random forests to resist over fitting is largely due to the number of trees and features, which is demonstrated in figure 1. Randomization increases bias but makes it possible to reduce the variance of the corresponding ensemble model through averaging. [11] It has been shown however that by increasing the maximum depth of each tree, a level of over fitting can be present, and similarly, that having shallow trees can produce low-confidence predictions. [12] It is thus important that an appropriate value of the maximum depth be used, as its optimal value is related to the problem complexity. We investigated the depth of trees on accuracy, shown in figure 2 below. We notice a small decrease in the accuracy at 20 features, and therefore choose an optimal at 15. This demonstrates that Random forests are still susceptible to over fitting, albeit less sensitive than many other machine learning algorithms without active regularization.

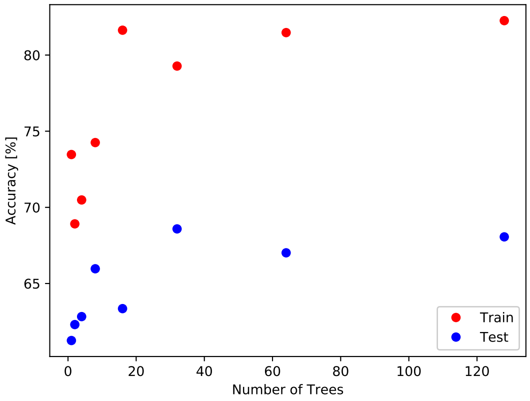

Figure 1: Showing Random forest inhibition to produce over fitting errors even with increased complexity by large number of trees.

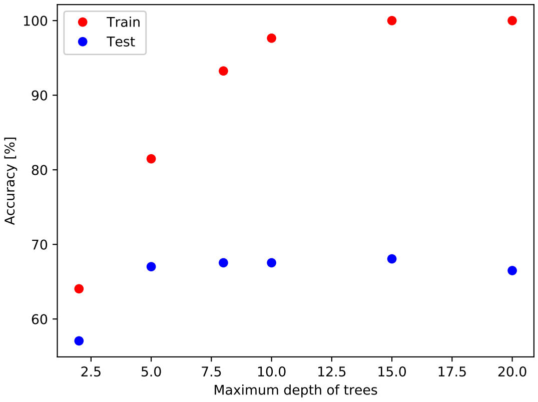

Figure 2: Showing accuracy of Random forest with 64 trees with varying maximum depth of each decision tree. It should be noted that there is a slight decrease in the test accuracy at 20 features which can possibly be related to over fitting on the training data which reaches 100% after 15.

From Table 2, it can be seen that all three implementations of Random forest yield similar results. It was noted however that when performing on standardized data so values are centered by their means and divided by the standard deviation, the accuracy was significantly lower for all three implementations, around 55% accuracy. This can be due to the sensitivity of the splitting using gini impurity or entropy with small values, or perhaps some information is lost on the impact of certain features. The time taken for completion is difficult to compare since coding implementations are different, however using 4 workers running using the same parameters as shown in table 1 the time of 3 minutes is slow, however not unreasonable.

Random forests have been shown to achieve high classification performance through ensemble with a set of decision trees that are constructed using randomly selected feature subspaces. The performance of an ensemble learner is dependent on the accuracy of each component learner and the diversity of the components, especially when using a small set of trees which may be limited due to computational cost. The randomization can cause occurrence of bad predicting trees as well as correlated trees which can lead to poor ensemble decisions, which can be observed when performing multiple training runs using the same parameters which can lead to different accuracy results.

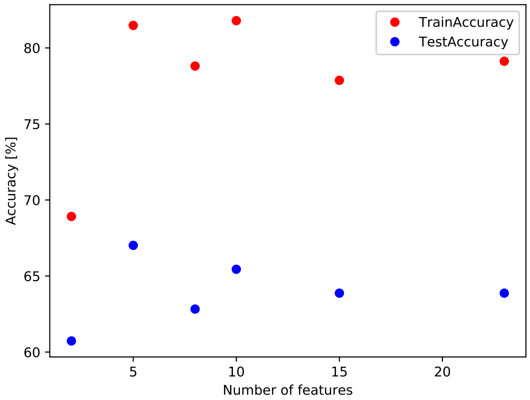

Figure 3: Showing the impact of the number of features used when constructing decision trees in Random forest with 64 trees and max depth 5. The optimal for testing is shown to be 5, which also agrees with literature on square-root of number of features

Table 1: Optimal values found for sci-kit learn Random forest by grid search

Target class	Max depth	Num features	Min samples per leaf	Min samples for split	Num of trees
GP_greater_than_0	8	5	2	2	128
sum_7yr_GP	8	5	4	2	64

Attempts have been made to improve the performance of this model by building a forest of only uncorrelated high performing trees. [13]

4 Conclusion

TODO: Add content.

Contributions

All authors contributed equally.
See GitLab project here for specific commits:
https://csil-git1.cs.surrey.sfu.ca/rkm3/mlclass-1777-randomforest

References

[1] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[2] T. Ho, “Recognition of handwritten digits by combining independent learning vector quantizations,” *Proceedings of the Second International Conference on Document Analysis and Recognition*, pp. 818–821, 1993.

[3] T. Ho, “Random decision forests,” *Proceedings of the Third International Conference on Document Analysis and Recognition*, pp. 278–282, 1995.

[4] T. Ho, “The random subspace method for constructing decision forests,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 832–844, 1998.

Table 2: Comparison of Random forest classifier for GP_greater_than_0

Machine Learning Package	**Accuracy**	**Time**
Weka	69.4%	<1sec
scikit-learn	70.7%	TIME
our implementation	68.6%	≈ 3min

[5] D. R. Cutler, T. C. Edwards, K. H. Beard, A. Cutler, T. Kyle, J. Gibson, J. J. Lawler, H. Beard, and T. Hess, "Random Forests for Classification in Ecology," *Ecology*, vol. 88, no. 11, pp. 2783–2792, 2007.

[6] J. R. Harris and E. C. Grunsky, "Predictive lithological mapping of Canada's North using Random Forest classification applied to geophysical and geochemical data," *Computers and Geosciences*, vol. 80, pp. 9–25, 2015.

[7] G. Luo and K. Wang, "A combined random forest and active contour-model approach to fully automatic segmentation of the left atrium in volumetric MRI," *BioMed Research International*, vol. 2017, 2017.

[8] N. Ghatasheh, "Business Analytics using Random Forest Trees for Credit Risk Prediction: A Comparison Study," *International Journal of Advanced Science and Technology*, vol. 72, pp. 19–30, 2014.

[9] D. Lock and D. Nettleton, "Using random forests to estimate win probability before each play of an NFL game," *Journal of Quantitative Analysis in Sports*, vol. 10, no. 2, pp. 197–205, 2014.

[10] J. Aslam, R. Popa, and R. Rives, "On estimating the size and confidence of a statistical audit," *Proc. Usenix/Accurate Electronic Voting Technology on Usenix/Accurate Electronic Voting Technology Workshop*, p. 8, 2007.

[11] S. Formann-Roe, "Bias and variance," Jun 2012.

[12] A. Criminisi, J. Shotton, and E. Konukoglu, "Decision Forests for Classification , Regression , Density Estimation , Manifold Learning and Semi-Supervised Learning," *Microsoft Research technical report*, vol. 7, pp. 81–227, 2011.

[13] S. Bharathidason and J. C. Venkataeswaran, "Improving Classification Accuracy based on Random Forest Model with Uncorrelated High Performing Trees," *International Journal of Computer Applications*, vol. 101, no. 13, pp. 26–30, 2014.