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 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. [1, 2, 3] The application of this method in his '93 paper to decision trees marked the introduction of Random Forests to the community. [1]

1.1 Decision Trees

TODO: Adjust titles as necessary and add content specifying other papers.

1.2 Ensemble Learning

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. [4]

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.

1.3 Random Forests

TODO: Adjust titles as necessary and add content specifying other papers.

2 Approach

TODO: Figure out what is supposed to be here.

3 Experiments

TODO: Add content.

3.1 Forest Size

TODO: Add content.

3.2 Tree Depth

TODO: Add content.

3.3 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. [3] 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. [3]

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.

Table 1: TODO: Use this table as a template for ours.

Machine Learning Package	DESCRIPTION
Weka	compare speed and accuracy at least
scikit-learn	compare speed and accuracy at least
Tensorflow(?) Probably need one more, but not sure which one yet.	compare speed and accuracy at least

3.3.1 Entropy

TODO: Add content.

3.3.2 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:

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

3.4 Custom Improvement?

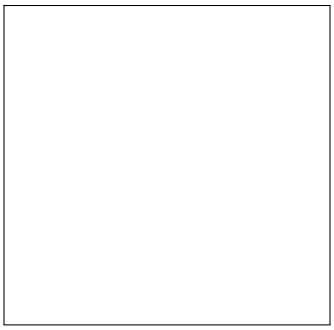

Figure 1: TODO: Use this figure as a template for ours.

Table 1.

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] 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.

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

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

[4] 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.

NIPS style guide:

References follow the acknowledgments. Use unnumbered third level heading for

the references. **Any choice of citation style is acceptable as long as you are**

consistent.

Example:

[1] Alexander, J.A. & Mozer, M.C. (1995) Template-based algorithms for connectionist rule extraction. In G. Tesauro, D. S. Touretzky and T.K. Leen (eds.), *Advances in Neural Information Processing Systems 7*, pp. 609-616. Cambridge, MA: MIT Press.