Bag of Little Random Forests (BLRF):
Adapting Random Forests to Big Data and Parallel Processing

Author: Zihao Xu Advisor: Dr. Johanna Hardin

ABSTRACT

Random Forests [1] are a successful ensemble method that utilizes a number of decision trees [2] to make predictions robust in both regression and classification settings [3]. However, the process of bootstrap aggregation, the mechanism underlying the random forest algorithm, requires each decision tree to physically store and perform computations on data sets of the same size as the input training set, a situation that is oftentimes impractical given the humongous sizes of data sets today. To address this problem, we introduce the Bag of Little Random Forests (BLRF), a new algorithm that adapts the Bags of Little Bootstraps [5], aiming to achieve a better computational profile while producing predictions with comparable accuracy as those of the standard random forest.

BIBLIOGRAPHY

[1] [Breiman, 2001] Breiman, L. (1 October, 2001). Random forests. Machine learning, Springer Netherlands, 45:5–32.
[2] [Bradley Efron, 1994] Bradley Efron, R. T. (May 15, 1994). An Introduction to the Bootstrap. CRC Press, illustrated, reprint edition.
[3] [Caruana et al., 2008] Caruana, R., Karampatziakis, N., and Yessenalina, A. (2008). An empirical evaluation of supervised learning in high dimensions. In Cohen, W. W., Mccallum, A., and Roweis, S. T., editors, Proceedings of the 25th International Conference on Machine Learning (ICML-08), pages 96–103.
[4] [Kleiner et al., 2014] Ariel Kleiner, Ameet Talwalkar, P. S. . I. J. (September 2014). A scalable bootstrap for massive data. Journal of the Royal Statistical Society, Volume 76, Issue 4: Pages 795 - 816.
[5] [Leo Breiman, 1999] Leo Breiman, Jerome H Friedman, R. A. O. C. J. S. (May 1999). Classification and Regression Trees. CRC Press, New York.
[6] [Yenny Zhang, 2017] Yenny Zhang, Dr. Johanna Hardin. (2017). Integrating random forests into the bag of little bootstraps. Submitted to Pomona College in Partial Fulfillment of the Degree of Bachelor of Arts.
[7] Picture from: http://blog.yhat.com/posts/random-forests-in-python.html
[8] [Wang et al., 2014] Wang, H., Zhuang, F., and He, Q. (2014). Scalable bootstrap clustering for massive data. SNPD.

CONTACT

Zihao Xu
Pomona College '19
Email: zihao.xu@pomona.edu
Phone: (310)-962-6992
Website: https://github.com/zihaoxu

Motivation and Objectives

The Random Forests (RF) algorithm (Figure 1) is inefficient when handling Big Data:

- **Time**: significant time consumption;
- **Memory**: physical storage of big data sets;
- **Structure**: not well-adapted to a parallelism;
- **Unable** to load the entire data set into memory.

Bag of Little Bootstraps (BLB) (Figure 2), can be used to mitigate the problem:

- **Time**: amount of computation is reduced;
- **Memory**: less physical storage of data sets;
- **Structure**: easy parallelization for individual subsamples;
- **Able** to deal with big data sets

Aim to: build BLRF (Figure 3). Reduce computation time while maintaining prediction accuracy.

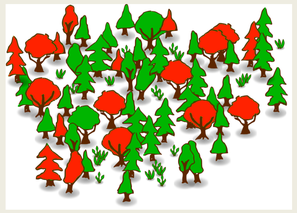

Figure 1 [7]. Visualization of RF

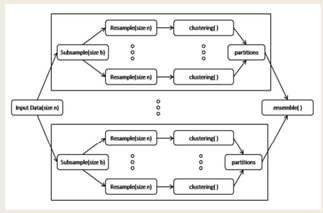

Figure 2 [8]. Visualization of BLB

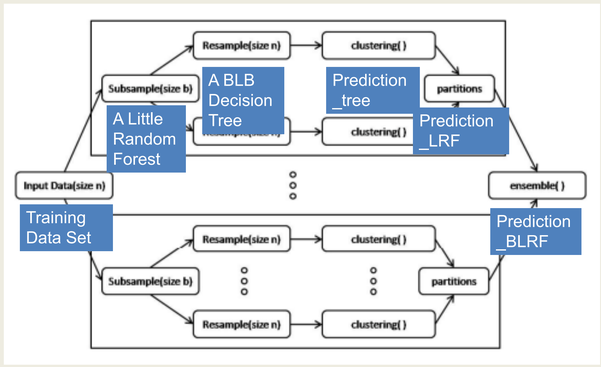

Figure 3. Visualization of BLRF

Steps and Procedures

- Study BLB and RF;
- Combine BLB and RF -> BLRF algorithm;
- Modify the source C code (regression only);
- Evaluate the BLRF algorithm (time and accuracy).

Table 1. Important BLRF parameters

Parameter	Definition in BLRF
n	Size of the training set
γ (gamma)	The user-defined parameter that determines value of ***b***, $b = n^{\gamma}$
b	Number of distinct observations in each Little Forest
s	Total number of Little Forests
ntree	Number of trees within each Little Forest

The BLRF Algorithm

Use "**BLB aggregation**" to replace the standard Bootstrap Aggregation. Given a training data set of size ***n***:

Pick a value for ***γ (gamma)***, calculate $b = n^{\gamma}$

↓

Draw ***s*** distinct subsamples of size ***b*** to build Little Random Forests

↓

Resample from each subsample, using the **multinomial method** (figure 4) w/ equal probabilities, to build BLB decision trees

↓

Average the results of BLB decision trees -> a Little Forest

↓

Average the results of Little Forest -> BLRF

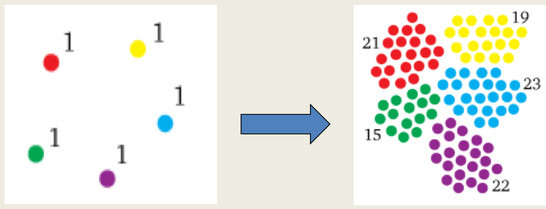

Figure 4 [6]. Illustration of the Multinomial Method:
b = 5, ***n*** = 100, p = {.2, .2, .2, .2, .2}, ***M*** = {21, 11, 22, 23, 19}

Conclusion and Discussion

What we have accomplished:
- Devised and implemented the BLRF algorithm (regression) in R;
- Produced results (Figure 8) comparable to those of random forests;
- Reduced time consumption (for ***γ < 0.9***).

Further directions:
- Fine-tune the parameters and find the optimal combination;
- Complete the classification part of BLRF;
- Further reduce computation time by implementing BLRF in C++.

BLRF Performance

Simulated Data - three data structures: ***n*** = 10000, ***ndim*** = 5
Linear: $Y = 5X_1 + 10X_2 + 15X_3 + 20X_4 + 25X_5 + \varepsilon$
Cosine: $Y = 50 \times \cos(\pi \times (X_1 + X_2)) + \varepsilon$
Clustered: $Y = cluster.means + \varepsilon$

Hardware - "Outlier", 64 AMD Opteron 6276 CPUs at 1.4 GHz

Statistical Performance - Figure 5 & 6 show the results of running the BLRF algorithm under different parameter settings:
- All measures of **RMSE** (**RMSE = MSE_{BLRF} / MSE_{RF}**) > 1, with those for ***γ*** = 0.9 close to 1, indicating relatively good results;
- **RMSE** decreases as ***s *, γ **** or ***ntree*** increases (*: larger effect);
- **RMSE** converges (△ <0.1%): ***s*** range: [3, 13], ***ntree*** range: [25,150]

Computational Performance – Figure 7 shows the results the *cosine* data set(holding ***ntree*** = 500), measured in seconds:
- Measures for time when ***γ*** is below 0.9 are smaller than the time require to build the random forest;
- Time goes up as ***s, γ*** (not shown) and ***ntree*** increases.

Figure 5. RMSE measures of BLRF algorithm: RMSE ~ s

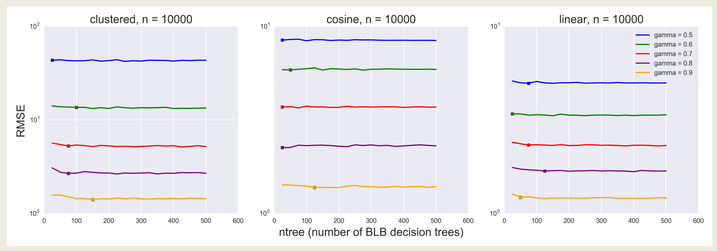

Figure 6. RMSE measures of BLRF algorithm: RMSE ~ ntree

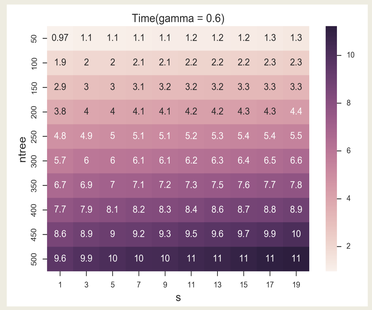

Figure 7. Time consumed to build the BLRF w/ different parameters

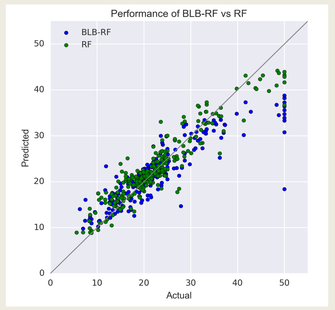

Figure 8. Performance of BLRF versus RF