Distribution Agreement

In presenting this thesis or dissertation as a partial fulfillment of the requirements for an advanced degree from Emory University, I hereby grant to Emory University and its agents the non-exclusive license to archive, make accessible, and display my thesis or dissertation in whole or in part in all forms of media, now or hereafter known, including display on the world wide web. I understand that I may select some access restrictions as part of the online submission of this thesis or dissertation. I retain all ownership rights to the copyright of the thesis or dissertation. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

Signature:

_____ _____

Haoyong Yu Date

Bagging for the highly adaptive lasso

By

Haoyong Yu
Master of Public Health

Biostatistics and Bioinformatics

David Benkeser, PhD
Thesis Advisor

Yi-an Ko, PhD
Reader

Bagging for the highly adaptive lasso

By

Haoyong Yu

B.E.
China Agricultural University
2017

Thesis Advisor: David Benkeser, PhD

An abstract of
A thesis submitted to the Faculty of the
Rollins School of Public Health of Emory University
in partial fulfillment of the requirements for the degree of
Master of Public Health
in Biostatistics and Bioinformatics
2020

Abstract

Bagging for the highly adaptive lasso
By Haoyong Yu

Prediction is a common goal in the statistic world. Many new estimators are created every year to seek better quality of prediction in various situation. A new estimator called highly adaptive lasso estimator was proved to be competitive with other popular machine learning methods and had theoretical advantages. Furthermore, the prediction performance of this estimator may be furthered by combining with some unique methods. Bagging is a common ensemble method that can be utilized to improve the performance of prediction. Feature bagging is a promising usage of traditional bagging method. We propose a new estimator that we call bagged highly adaptive lasso estimator based on feature bagging approach. We show via simulation and public data analysis that our estimator seems not provide more benefits by additional aggregating bootstrap procedures.

Bagging for the highly adaptive lasso

By

Haoyong Yu

B.E.
China Agricultural University
2017

Thesis Advisor: David Benkeser, PhD

A thesis submitted to the Faculty of the
Rollins School of Public Health of Emory University
in partial fulfillment of the requirements for the degree of
Master of Public Health
in Biostatistics and Bioinformatics
2020

Table of Contents

1. Introduction

Machine learning is currently one of the most significant and influential technologies around the world. In contrast to traditional data analysis, which combines relatively simple analytical approaches with data to find answers to a problem, machine learning uses complex computational algorithms to discover the rules behind a problem (Chollet, 2019). Uncovering underlying patterns in a problem can be useful for making predictions (Patel et al., 2015), recognizing images or speech (Hoang, 2018), making medical diagnoses (Lo & Jack Li, 2018), and investigating fraud (Awoyemi, Adetunmbi, & Oluwadare, 2017). Machine learning approaches are diverse and often rely on classic computational algorithms such as the decades-old Expectation Maximization (EM) algorithm (Dempster, Laird, & Rubin, 1977), Principal Component Analysis (Jolliffe, 1986), the Support Vector Machine (SVM) algorithm (Cortes & Vapnik, 1995), the Adaboost algorithm (Freund & Schapire, 1996), the Random Forests algorithm (Breiman, 2001), the modern Deep Convolutional Neural Networks algorithm (Krizhevsky, Sutskever, & Hinton, 2012), among others.

Prediction performance is often an important criterion for assessing a model or an algorithm (Vihinen, 2012). Ensemble methods help produce better predictive performance by combining multiple predictions from one or several machine learning algorithms. Bootstrap Aggregation (Bagging) is a one of the most widely used ensemble methods, which can be used to reduce the variance of many algorithms that often exhibit prohibitively high variance and to avoid overfitting. Bagging involves taking a bootstrap sample of the data, training the predictive model on each bootstrap sample, and obtaining the final model by sensibly averaging each bootstrapped model. For example, bagging was found to reduce the misclassification rates of

classification and regression trees by at least 20% and 22%, respectively (Breiman, 1996). In order to reduce the correlations between each estimator, feature bagging was created by training models on random features rather than the entire set of features (Lazarevic & Kumar, 2005). Furthermore, feature bagging was seen to further improve performance, leading to the now-famous Random Forests algorithm.

Highly Adaptive Lasso (HAL) is a new approach to machine learning that has been shown to have several theoretical advantages (Benkeser & Van Der Laan, 2016). The approach does not require local smoothness assumptions (e.g., differentiability of the underlying function that is estimated) and the convergence rate of the performance of the HAL estimator relative to that of the optimal prediction function converges faster than $n^{-1/4}$ even in high dimensions. HAL has been shown to have competitive prediction performance compared to other popular machine learning algorithms in real and simulated data. Even still, it may be possible to further improve the performance of HAL by coupling it with bagging. In this work, we investigate whether this is indeed the case. We propose a computationally efficient approach to bagging the HAL estimator and explore using real and simulated data whether the approach leads to improvements in predictive performance.

2. Methods

2.1 Highly adaptive lasso

The highly adaptive lasso estimator is a nonparametric regression estimator (Benkeser & Van Der Laan, 2016). The method relies on relatively mild smoothness assumptions, while achieving a fast convergence rate to the true regression function. To implement a HAL of the

regression of a continuous-valued outcome *Y* on covariates *X*, one first generates a set of basis functions consisting of indicators of each data point. For example, if *X* is univariate and we have *n* observed data points, then the basis functions are $\phi_{\mathrm{k}}(\mathrm{x}) = \mathrm{I}(\mathrm{x} \geq \mathrm{X_k})$, for $\mathrm{k} = 1, \ldots, \mathrm{n}$. Thus, the regression function is

$$E(Y \mid X = x) = \beta_0 + \beta_1 \phi_{\mathrm{i1}}(\mathrm{x}) + \beta_2 \phi_{\mathrm{i2}}(\mathrm{x}) + \cdots + \beta_n \phi_{\mathrm{in}}(\mathrm{x}).$$

For each $s > 0$, the regression coefficients are estimated by

$$argmin_{\beta} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{k=1}^{p} \beta_k \phi_{\mathrm{ik}}(\mathrm{x_i})\right)^2 \text{ subject to } \sum_{k=1}^{p} |\beta_k| \leq s \ ,$$

where s is the L_1-norm of the coefficient vector. The value of s is selected using ten-fold cross-validation.

In the bivariate setting where $X = (X_1, X_2)$, first-order basis functions are generated as above for $\mathrm{X_1}$ and $\mathrm{X_2}$ and second-order basis expansion functions are also created as $\phi_{12,\mathrm{k}}(\mathrm{x}) = \mathrm{I}(\mathrm{x_1} \geq \mathrm{X_{1k}}, \mathrm{x_2} \geq \mathrm{X_{2k}})$ for $\mathrm{k} = 1, \ldots, \mathrm{n}$.

2.2 Bagging

Bootstrap Aggregation (Bagging) is a machine learning ensemble method to reduce variance and improve stability of an estimated prediction function. It mostly helps high-variance, low-bias classifiers. The core procedures in bagging are as follows. Suppose we have dataset *Z*, where $\mathrm{Z} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$. We first generate a bootstrap data set by sampling with replacement from *Z*. For each bootstrap dataset $Z^* = \{(x_1^*, y_1^*), (x_2^*, y_2^*), \ldots, (x_n^*, y_n^*)\}$, we can build a prediction model $\hat{f}^{*b}(x)$. With *B* models in total, the final model is obtained by averaging the predictions, $\hat{f}_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B} \hat{f}^{*b}(x)$.

Another usage of bagging is so-called *feature bagging*. Here, we draw a random subset of

covariates from training dataset for the multivariate setting, where the number of random covariates to select is a user-selected tuning parameter. For example, when X is three-dimensional, but we are using feature bagging to only sample two covariates, then if the original dataset is $Z = \{(x_{11}, x_{12}, x_{13}, y_1), (x_{21}, x_{22}, x_{23}, y_2), \dots, (x_{n1}, x_{n2}, x_{n3}, y_n)\}$, a bootstrap data set with feature bagging could be $Z^* = \{(x_{11}^*, x_{13}^*, y_1^*), (x_{21}^*, x_{23}^*, y_2^*), \dots, (x_{n1}^*, x_{n3}^*, y_n^*)\}$, where in this sample the first and third covariates were re-sampled. The final prediction model is obtained by averaging over repeated fits.

2.3 Bagged highly adaptive LASSO

We propose a bagged HAL method and investigate whether bagging improves predictions over a regular HAL. Our proposed algorithm for generating bagged HAL predictions is as follows. First, we generate a bootstrap data set using feature bagging as described above. For each bootstrap sample, we refer to the observations not included in the sample as the *out-of-bag (OOB)* observations. Using each bootstrap data set, a HAL model is fit. That is, we obtain a solution along the LASSO path for the regression function based on the bootstrap data. To select the L-1 bound on the coefficient vector, we use the OOB observations to compute a measure of predictive accuracy. The L-1 penalty that maximizes OOB predictive accuracy is selected, thus generating the *b*-th bagged HAL fit. The process is repeated *n*HAL times, where *n*HAL is a user-selected tuning parameter. The final bagged HAL estimator results from averaging the *n*HAL different fits.

3. Simulation

We evaluated the prediction performance of bagged *n*HAL estimators with $nHAL \in \{10, 20,$

$50, 100\}$ compared to single HAL in four data generating scenarios. The outcomes of first two scenarios are continuous, while the latter two are binary. We changed the dimension of X (d) and $\mathrm{d} \in \{3,7\}$ for each type of outcome. In dimension d = 3, the distributions of $\mathrm{X}_1, \mathrm{X}_2$ and X_3 and the regression function were as follows:

$$\mathrm{X}_1 \sim \mathrm{Uniform}(0,1); \mathrm{X}_2 \sim \mathrm{Normal}(30,5); \mathrm{X}_3 \sim \mathrm{Normal}(3,0.5)$$

$$\mathrm{Y} = 3\mathrm{x}_1 - 0.3\mathrm{x}_2 + \varepsilon \text{ where } \epsilon \sim \mathrm{Normal}(0,0.5) \text{ for continuous outcome}$$

$$\text{Logit } Y = 3\mathrm{x}_1 - 0.1\mathrm{x}_2 - 0.5 \text{ for binary outcome}$$

If dimension d = 7, the regression was created as follows:

$$\mathrm{X}_1 \sim \mathrm{Bernoulli}(0.3); \mathrm{X}_2 \sim \mathrm{Uniform}(-3,3); \mathrm{X}_3 \sim \mathrm{Normal}(30,0.5)$$

$$\mathrm{X}_4 \sim \mathrm{Normal}(3,0.5); \mathrm{X}_5 \sim \mathrm{Bernoulli}(0.7); \mathrm{X}_6 \sim \mathrm{X}_5 \sim \mathrm{Bernoulli}(0.2)$$

$$\mathrm{X}_7 \sim \mathrm{Poisson}(2).$$

$$\mathrm{Y} = 5\mathrm{x}_1 - 0.5\mathrm{x}_2 + \varepsilon \text{ where } \epsilon \sim \mathrm{Normal}(0,0.5) \text{ for continuous outcome}$$

$$\text{Logit } Y = 5\mathrm{x}_1 - 0.5\mathrm{x}_2 + 0.5 \text{ for binary outcome}$$

For each scenario, only two features are predictive of the outcome, while the others are noise. We considered training sets of size $l \in \{50, 100, 200, 500, 1000\}$. Prediction performance was assessed according to MSE or AUC, which is computed on another independent testing set with $m = 5000$ observations. The whole simulation procedure was repeated 1000 times.

The results of scenarios with binary outcomes are displayed in Figure 1. The boxplots in each row represent MSE in different scenarios and the boxplots in each column represent MSE of HAL and Bagged HAL with different n. The bottom margin displays the overall results of MSE

of HAL and Bagged HAL across two scenarios combined. The lower the MSE values, the better the prediction performance. We found that HAL performs considerably better than bagged HAL in all cases in light of the MSE and stability. For the bagged HAL, as the *n* enlarges, both the MSE and stability of prediction performance grows. However, the difference between HAL and bagged HAL doesn't change with sample size increasing in training sets. The results of scenarios 2 and the overall results show the same thing. But note that the stability decreases as dimension of X increases. In a nutshell, HAL performs better than the bagged HAL in regression functions with continuous outcome.

Figure 2 shows the results of latter two scenarios with binary outcomes in the simulation study. The format of Figure 2 is the same as that of Figure 1, except the vertical axis in each boxplot represent AUC rather than MSE. The higher the AUC values, the better the prediction performance. From the top row, we can notice that when the sample size of training set is small ($l = 100$), the Bagged HAL performs a little better than single HAL in terms of AUC and stability. However, as the sample size increases to 200, the prediction performance of HAL jumps to the highest and keeps a high-level performance with sample size growing. But when the sample size of training set is large enough, HAL and bagged HAL perform comparably. Meanwhile, the performance of bagged HAL is gradually improved as *n* increases. The growth trend for bagged HAL in higher dimension is similar to that in scenario 3. However, the performance of HAL is much better than that of bagged HAL in scenario 4 no matter how much the sample size of training set is. Across all the dimensions and sample sizes, the single HAL has the best overall performance. To sum up, HAL performs better than bagged HAL in the setting of binary outcome, but bagged HAL take the lead when the sample size of training set

is small in low dimensions.

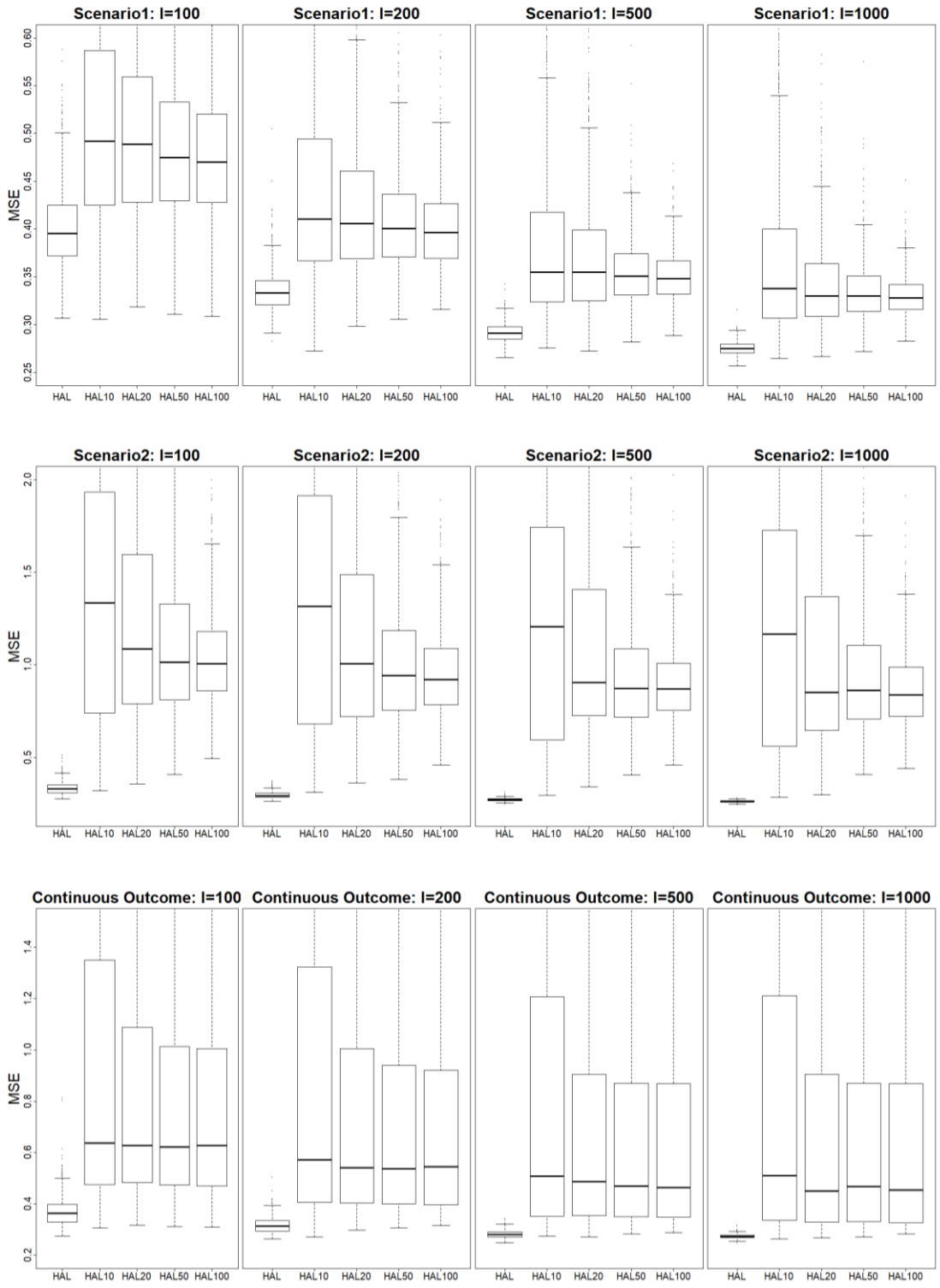

Figure 1 Simulation study results for continuous outcomes

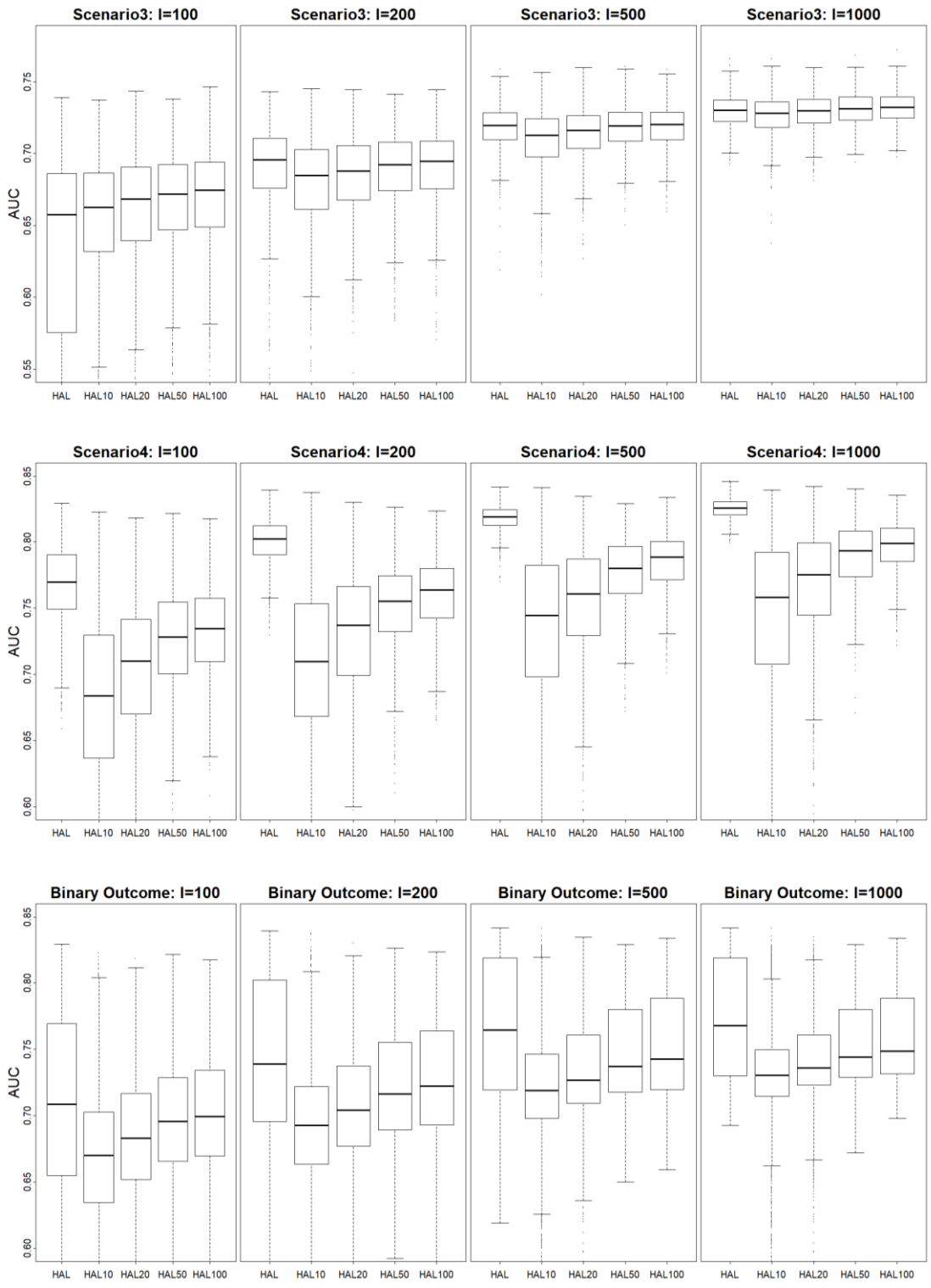

Figure 2 Simulation study results for binary outcomes

4. Data Analysis

We are interested in the prediction performance of Bagged HAL in real data examples, so we analyzed two publicly available data sets, *wine* and *drugs* (Benkeser, Petersen, & van der Laan, 2019). The outcome of these two data sets are both binary. The sample sizes range from 1885 to 6497 and each dataset has 12 covariates. In addition to the HAL and bagged HAL estimators, we considered traditional machine learning algorithm including Random Forest and Gradient Boosting as well. The split ratio of training set to testing set is 0.7. For the bagged HAL estimator, we randomly picked 5 features for *wine* and *drugs*, respectively. The average AUC over 100 repeated simulations was reported as follows.

Table 1 AUC of four methods in two real data sets

	HAL	Bagged HAL nHAL=10	Bagged HAL nHAL=20	Bagged HAL nHAL=50	Bagged HAL nHAL=100	Random Forest	Gradient Boosting
Wine							
	0.87002	0.86380	0.86588	0.87080	0.87306	0.91287	0.89043
Drugs							
	0.74623	0.74333	0.74294	0.74620	0.74590	0.72766	0.70655

The prediction performance for the four algorithms regarding AUC on testing set are shown in Table 1. For the wine dataset, Random Forest and Gradient Boosting had better performance than HAL family; however, HAL and bagged HAL performed acceptably. In the drugs data set, HAL had the highest performance, though there was little to distinguish its performance from Bagged HAL or random forest. Overall, we can find that the performance of HAL is basically equivalent with that of Bagged HAL for these two datasets and increasing nHAL of bagged HAL does not affect the results much.

5. Conclusion

In this project, we proposed a new highly adaptive lasso estimator based on bagging method. We assessed the prediction performance of this bagged HAL using both simulations and public data to investigate whether bagging improves predictions over a regular HAL.

The results of simulations show that the overall performance of single HAL is better than bagged HAL, particularly for predicting continuous outcomes. Meanwhile, the stability of bagged HAL improves is not as fast as that of HAL regardless of the dimensions. Computational problems are more severe in bagged HAL, which is caused by the bootstrap step. Therefore, the bagged HAL appears not be an efficient estimator compared to HAL. However, in low dimensions, when the sample size of training set is small, the performance of bagged HAL is reliable, and we could further the investigation.

The results of public data analysis display that both HAL and Bagged HAL are competitive methods comparing with traditional machine learning method, especially when the sample size of training set is limited. However, for these examples, we again found little difference between bagged HAL and regular HAL.

Our analysis has several limitations. First, the results of our simulations are based on moderate sample sizes. In the future, it may be of interest to compare the performance in larger training samples. Second, a more comprehensive assessment of how sparsity affects performance may also be of interest. Because of the feature bagging step, bagged HAL only ever includes a subset of covariates in a given HAL fit. Thus, we may expect the relative performance to improve in more sparse settings. Finally, bagging has been noted to be successful in stabilizing the

performance of highly variable machine learning techniques that are prone to overfitting, such as regression trees. HAL may be too stable of an algorithm on its own to see much benefit from bagging. Therefore, it may be of interest to try a different approach, where for each bagged HAL fit we select L-1 norm of HAL *larger* than the one recommended based on OOB performance. In this case, the bias of the resultant HAL fit will decrease, bust variance will increase. However, by aggregating over many bagged HALs we may appropriately decrease the variance.

In conclusion, our study found that bagging does not significantly improve the performance of HAL. Future studies may identify different approaches or scenarios to further improve HAL performances.

References

Awoyemi, J. O., Adetunmbi, A. O., & Oluwadare, S. A. (2017, 29-31 Oct. 2017). *Credit card fraud detection using machine learning techniques: A comparative analysis.* Paper presented at the 2017 International Conference on Computing Networking and Informatics (ICCNI).

Benkeser, D., Petersen, M., & van der Laan, M. J. (2019). Improved Small-Sample Estimation of Nonlinear Cross-Validated Prediction Metrics. *Journal of the American Statistical Association*, 1-16. doi:10.1080/01621459.2019.1668794

Benkeser, D., & Van Der Laan, M. (2016). *The highly adaptive lasso estimator.* Paper presented at the 2016 IEEE international conference on data science and advanced analytics (DSAA).

Breiman, L. (1996). Bagging predictors. *Machine learning, 24*(2), 123-140.

Breiman, L. (2001). Random forests. *Machine learning, 45*(1), 5-32.

Chollet, F. (2019). *Deep learning with Python*.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning, 20*(3), 273-297.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological), 39*(1), 1-22.

Freund, Y., & Schapire, R. E. (1996). *Schapire R: Experiments with a new boosting algorithm.* Paper presented at the In: Thirteenth International Conference on ML.

Hoang, N. D. (2018). Image Processing-Based Recognition of Wall Defects Using Machine Learning Approaches and Steerable Filters. *Comput Intell Neurosci, 2018*, 7913952. doi:10.1155/2018/7913952

Jolliffe, I. T. (1986). Principal components in regression analysis. In *Principal component analysis* (pp. 129-155): Springer.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). *Imagenet classification with deep convolutional neural networks.* Paper presented at the Advances in Neural Information Processing Systems.

Lazarevic, A., & Kumar, V. (2005). *Feature bagging for outlier detection.* Paper presented at the Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining.

Lo, C. M., & Jack Li, Y. C. (2018). The use of multimedia medical data and machine learning for various diagnoses. *Comput Methods Programs Biomed, 165*, A1. doi:10.1016/j.cmpb.2018.09.008

Patel, M. J., Andreescu, C., Price, J. C., Edelman, K. L., Reynolds, C. F., 3rd, & Aizenstein, H. J. (2015). Machine learning approaches for integrating clinical and imaging features in late-life depression classification and response prediction. *Int J Geriatr Psychiatry, 30*(10), 1056-1067. doi:10.1002/gps.4262

Vihinen, M. (2012). *How to evaluate performance of prediction methods? Measures and their interpretation in variation effect analysis.* Paper presented at the BMC genomics.