**MATH 569: Statistical Learning**

Applied Mathematics Department
Illinois Institute of Technology

---

# BAGGING PREDICTORS

Harshaa Bajaj, Sarath Tamarana, Shiva Naveen Ravi
(A20378813),(A20365712),(A20360877)

---

**Instructor:** Prof. Lulu Kang

# Index

# 1. Abstract

The goal of ensemble methods is to combine the predictions of several base estimators built with a given learning algorithm in order to improve generalizability/robustness over a single estimator. Two families of ensemble methods are usually distinguished:

- In averaging methods, the driving principle is to build several estimators independently and then to average their predictions. On average, the combined estimator is usually better than any of the single base estimator because its variance is reduced.[1]
- Plurality voting (PV), where the class with the most votes wins. It is one form of majority voting.

In ensemble algorithms, bagging methods form a class of algorithms which build several instances of a black-box estimator on random subsets of the original training set and then aggregate (either by voting or by averaging) their individual predictions to form a final prediction. These methods are used as a way to reduce the variance of a base estimator by introducing randomization into its construction procedure and then making an ensemble out of it. In many cases, bagging methods constitute a very simple way to improve with respect to a single model, without making it necessary to adapt the underlying base algorithm. As they provide a way to reduce overfitting, bagging methods work best with strong and complex models (e.g., fully developed decision trees), in contrast with boosting methods which usually work best with weak models (e.g., shallow decision trees).

The idea behind bagging is when the model overfits with a nonparametric regression method (usually regression or classification trees), it tends to have high variance, no (or low) bias, so we can take many resamples (from bootstrapping), each overfitting, and average them together. This should lead to the same bias (low) but cancel out some of the variance.

The purpose of this study is to improve our understanding on Bagging and implement the techniques used in Bagging Predictors by Leo Breiman.[2]

# 2. Introduction

Models which are combinations of other models are called an ensemble. In statistics and machine learning, ensemble methods use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone.[3]

A statistical ensemble is an idealization consisting of a large number of virtual copies (sometimes infinitely many) of a system, considered all at once, each of which represents a possible state that the real system might be in. In other words, a statistical ensemble is a probability distribution for the state of the system.[4]

Unlike a statistical ensemble in statistical mechanics, which is usually infinite, a machine learning ensemble refers only to a concrete finite set of alternative models, but typically allows for much more flexible structure to exist among those alternatives.

These methods are learning models that achieve performance by combining the opinions of multiple learners. In doing so, we can often get away with using much simpler learners and still achieve great performance.

Moreover, ensembles are inherently parallel, which can make them much more efficient at training and test time, if we have access to multiple processors.[5]

Generally meta algorithms consists of two steps:

1. Producing a distribution of simple ML models on subsets of the original data.

2. Combining the distribution into one aggregated model.

There are three approaches to combine several machine learning techniques into one predictive model in order to decrease the variance (bagging), bias (boosting) or improve the predictive force (stacking alias ensemble).

## 2.1 Bagging

Bagging was proposed by Leo Breiman in 1994 to improve the classification by combining classifications of randomly generated training sets.[6]

The paper[7] being studied for this project was the first fundamental approach to introduce Bagging and prove that it reduces variance for unstable models.

Bagging (Bootstrap Aggregation) decreases the variance of predictions by generating additional data for training from the original dataset by producing multisets of the same cardinality/size. Although it is usually applied to decision tree methods, it can be used with any type of method. It is a special case of the model averaging approach.

Bagging methods come in many flavours but mostly differ from each other by the way they draw random subsets of the training set:

- When random subsets of the dataset are drawn as random subsets of the samples, then this algorithm is known as Pasting.
- When samples are drawn with replacement, then the method is known as Bagging.
- When random subsets of the dataset are drawn as random subsets of the features, then the method is known as Random Subspaces.
- Finally, when base estimators are built on subsets of both samples and features, then the method is known as Random Patches.[1]

## 2.1.1 Method

Given a standard training set D of size n, bagging generates m new training sets $D_i$ , each of size n′, by sampling from D. The m models are fitted using the above m samples and combined by averaging the output (for regression) or voting (for classification).

In other words,

- For each iteration t where t=1,…T,
    - N random samples with replacement are selected from the training set
    - A chosen model is then applied on the samples.
- For each test example
    - Start all trained base models
    - Predict by combining results of all T trained models:
        - Regression: averaging
        - Classification: a majority vote

To summarize,

| | Bagging |
|---|---|
| Partitioning of the data into subsets | Random |
| Goal to achieve | Minimize variance |
| Methods where this is used | Random subspace |
| Function to combine single models | (Weighted) average |

## 2.1.3 Advantages

- Bagging leads to improvements for unstable procedures, which include, for example, artificial neural networks, classification and regression trees, and subset selection in linear regression.
- On the other hand, it can mildly degrade the performance of stable methods such as K-nearest neighbors.[6]
- Parallel ensemble: each model is built independently
- Aim to decrease variance, not bias
- Suitable for high variance low bias models (complex models)[8]

# 3. Bagging Classification Trees

Bagging was applied to classification trees of the breast cancer, glass, ionosphere and waveform (simulated) data sets

The table below gives a numerical summary of the data sets

| Data Set Summary | | | |
|---|---|---|---|
| Data Set | # Samples | # Variables | # Classes |
| waveform | 1800 | 21 | 3 |
| breast cancer | 699 | 9 | 2 |
| ionosphere | 351 | 34 | 2 |
| glass | 214 | 9 | 6 |

Procedure to implement bagging:

1. The data set is randomly divided with respect to 90:10 train test split. For the waveform data set, from the training set 1500 records and from the test set 300 records are sampled out for generating new training and test sets to work with.
2. A classification tree is constructed from the training set using 10-fold cross-validation.
3. The model is run on the test set and misclassification error rate ($e_S$) is calculated

4. Bagging is applied to the training set forming classification trees using single tree classifiers on 50 bootstrap samples as follows,

    a. A new classification tree is constructed from each bootstrap sample using 10-fold cross-validation.

    b. The model is then run on the test set where the estimated class of $x_n$ is the class that has the majority votes in the previous classifiers. If there is a tie, the estimated class is the one with the lowest class label.

5. The misclassification error rate ($e_B$) is the proportion of times the estimated class differs from the true class

6. The random division of the data into learning and training sets is repeated 100 times and the reported $\bar{e}_S$ , $\bar{e}_B$ are the averages over the 100 iterations.

The table below shows the values of $\bar{e}_S$ and $\bar{e}_B$

| **Misclassification Rates (%)** | | | |
|---|---|---|---|
| **Data Set** | $\bar{e}_S$ | $\bar{e}_B$ | **Decrease** |
| waveform | 24.79 | 17.07 | 31% |
| breast cancer | 7.57 | 4.46 | 41% |
| ionosphere | 11.50 | 9.25 | 20% |
| glass | 37.60 | 29.67 | 21% |

The table below shows the estimated standard errors

| Standard Errors of Misclassification | | |
|---|---|---|
| **Data Set** | **SE($\bar{e}_S$)** | **SE($\bar{e}_B$)** |
| waveform | 0.23 | 0.13 |
| breast cancer | 0.31 | 0.24 |
| ionosphere | 0.52 | 0.47 |
| glass | 1.18 | 0.91 |

# 3.1 Theoretical Justification

In classification, a predictor $\phi(x,L)$ predicts a class label $j \in \{1,...,J\}$.

Let,

$$Q(j|x) = P(\varphi(x,L) = j)$$

i.e, over many independent replicates of the learning set L, φ predicts class label j at input x with relative frequency Q( j|x )

Let P(j|x) be the probability that input x generates class j. Then the probability that the predictor classifies the generated state at x correctly is

$$\Sigma_j Q(j|x) P(j|x)$$

The overall probability of correct classification is

$$r = \int [\Sigma_j Q(j|x) P(j|x)] P_X(dx)$$

where $P_X(dx)$ is the x probability distribution

For any Q(j|x),

$$\Sigma_j Q(j|x) P(j|x) \leq \max_j P(j|x), \text{ with equality only if}$$

$$Q(j|\boldsymbol{x}) = \begin{cases} 1 & \text{if } P(j|\boldsymbol{x}) = \max_i P(i|\boldsymbol{x}) \\ 0 & \text{else} \end{cases}$$

The aggregated predictor is: $\varphi_A(x) = \text{argmax}_j Q(j|x)$.

For the aggregated predictor the probability of correct classification at x is

$$\Sigma_j I\ (\text{argmax}_i\, Q(i|x) = j\ )\ P(\ j|x\ )$$

where I is the indicator function.

If φ is order-correct at x, then the above equals $\max_j P(\ j|x\ )$.

Assuming C to be the set of all inputs x at which φ is order-correct, we get for the correct classification probability of $\varphi_A$ the expression

$$r_A = \int_{x\in C} \max_j P(\ j|x\ )\ P_X(dx) + \int_{x\in C'} [\ \Sigma_j\ I\ (\ \varphi_A(x) = j\ )\ P(\ j|x\ )\ ]\ P_X(x)$$

Even if φ is order-correct at x its correct classification rate can be far from optimal. But $\varphi_A$ is optimal. If a predictor is good in the sense that it is order-correct for most inputs x, then aggregation can transform it into a nearly optimal predictor.

On the other hand, unlike the numerical prediction situation, poor predictors can be transformed into worse ones. The same behavior regarding stability holds. Bagging unstable classifiers usually improves them. Bagging stable classifiers is not a good idea.

# 4. Bagging Regression Trees

Bagging trees was used on Boston Housing and the Friedman #1 data sets which have numerical responses.

A summary of these data sets is mentioned below,

| Data Set Summary | | | |
|---|---|---|---|
| **Data Set** | **# Samples** | **# Variables** | **# Test Set** |
| Boston Housing | 506 | 12 | 51 |
| Friedman #1 | 200 | 10 | 1000 |

Procedure to implement Bagging:

1. The data set is randomly divided into training and test set with respect to 90:10 train test split.
2. A regression tree is constructed from the training set using 10-fold cross-validation.
3. The model is then run on the test set and squared error ($e_S$) is calculated
4. Bagging is then applied to the training set forming regression trees using single regression trees on 25 bootstrap samples as follows,
    a. A new regression tree is constructed from each bootstrap sample using 10-fold cross-validation and pruned using the training set

b. The model is then run on the test set where for $(y_n, x_n) \in T$ , the bagged predictor is

$$\hat{y}_n = av_k \varphi_k(x_n)$$

and the squared error is

$$e_B = av_n (y_n - \hat{y}_n)^2$$

5. The random division of the data into learning and training sets is repeated 100 times and the reported $\bar{e}_S$ , $\bar{e}_B$ are the averages over the 100 iterations.

The table below lists the values of $\bar{e}_S$ and $\bar{e}_B$

| **Mean Squared Test set Error** | | | |
|---|---|---|---|
| **Data Set** | $\bar{e}_S$ | $\bar{e}_B$ | **Decrease** |
| Boston Housing | 20.33 | 14.11 | 31% |
| Friedman #1 | 11.42 | 7.22 | 37% |

Estimated standard errors:

| **Standard Errors** | | |
|---|---|---|
| **Data Set** | **SE($\bar{e}_S$)** | **SE($\bar{e}_B$)** |
| Boston Housing | 1.03 | 0.58 |
| Friedman #1 | 0.10 | 0.06 |

# 4.1 Theoretical Justification

Let each (y, x) case in L be independently drawn from the probability distribution P.

Suppose y is numerical and $\varphi(x,L)$ the predictor. Then the aggregated predictor is the average over L of $\varphi(x,L)$, i.e,

$$\varphi_A(x) = E_L\varphi(x,L)$$

Take Y, X to be random variables having the distribution P and independent of L. The average prediction error e in $\varphi(x,L)$ is

$$e = E_L E_{Y,X}( Y - \varphi(X,L) )^2$$

Define the error in the aggregated predictor $\varphi_A$ to be

$$e_A = E_{Y,X}(Y - \varphi_A(X,P))^2$$

Using the inequality $EZ^2 \geq (EZ)^2$ gives,

$$\begin{aligned} e &= EY^2 - 2EY\varphi_A + E_{Y,X}E_L\varphi^2(X,L) \\ &\geq E( Y - \varphi_A )^2 \\ &= e_A \end{aligned}$$

Thus, $\varphi_A$ has lower mean-squared prediction error than $\varphi$.

Assuming we measure a random variable x having normal distribution ($N(\mu,\sigma^2)$).

If only one measurement x1 is done,

- Expected mean of the measurement is $\mu$
- Variance is $Var(x1)=\sigma^2$

If random variable x is measured K times $(x_1,x_2,\ldots,x_k)$ and the value is estimated as: $(x_1,x_2,\ldots,x_k)/K$,

- Mean of the estimate is still $\mu$
- But, variance is smaller: $[Var(x_1)+\ldots+Var(x_k)]/K^2$

$$=K\sigma^2 / K^2$$

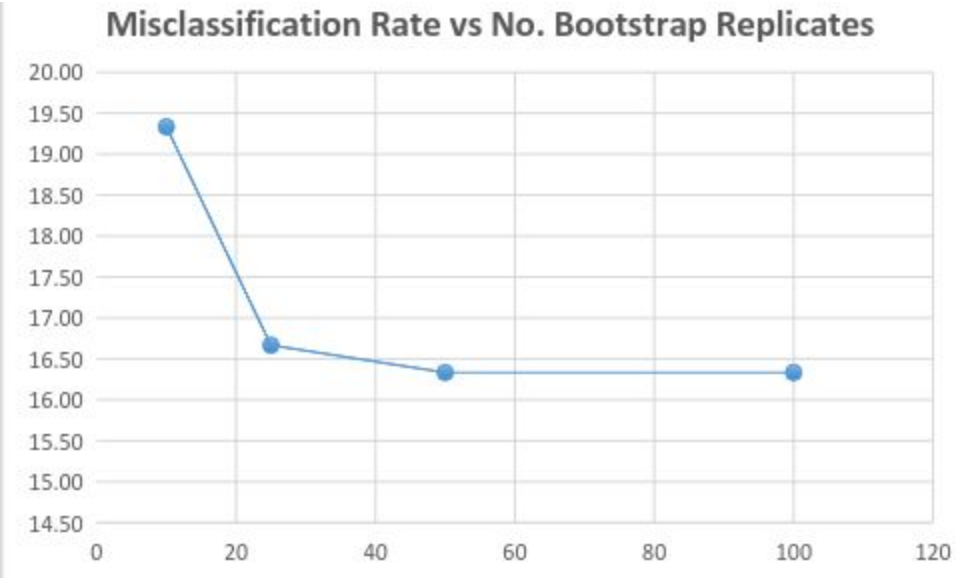
$$= \sigma^2/K$$

# 5. Number of Bootstraps

In the previous implementations we have used 50 and 25 replicates for classification and regression respectively and it is evident that bootstrapping improve the accuracy of the models but there does not seem to be a rule of thumb to decide the number of bootstrap samples to be used.

Just to get a sense of how the number of samples affect the accuracy we ran the waveform (simulated) data using 10, 25, 50 and 100 replicates and implemented the same bagging technique as mentioned previously. The following are the results obtained

| **Bagged Misclassification Rates %** | |
|---|---|
| **No. Bootstrap Replicates** | **Misclassification Rate** |
| 10 | 19.33 |
| 25 | 16.67 |
| 50 | 16.33 |
| 100 | 16.33 |




Clearly until 25 samples the misclassification rate significantly decreases and then there does not seem to be much difference

# 6. Conclusion

From the tests on data sets using classification and regression trees, where in both cases the size of the single decision tree and of the bootstrapped trees was chosen by optimizing a 10-fold cross-validated error, and the theoretical evidence from the reference paper it was patent that bagging can give substantial gains in accuracy.

Following the fact that a linear combination of trees is much harder to interpret than a single tree becomes the main disadvantage of bagging, its lack of interpretability.

# 7. Appendix

Descriptions of Data Sets

1. Classification Data Sets

### 1.1. Breast Cancer

It is two class data with 699 cases (458 benign and 241 malignant). It has 9 variables consisting of cellular characteristics

### 1.2. Ionosphere

There are 351 cases with 34 variables, consisting of 2 attributes for each at 17 pulse numbers. There are two classes: good = some type of structure in the ionosphere (226); bad = no structure (125).

### 1.3. Glass Identification

Each case consists of 9 chemical measurements on one of 6 types of glass. There are 214 cases.

### 1.4. Waveform

This is simulated 21 variable data with 300 cases and 3 classes each having probability ⅓. The dataset was downloaded from the UCI repository.

## 2. Regression Data Sets

### 2.1. Boston Housing

It has 506 cases corresponding to census tracts in the greater Boston area. The y-variable is median housing price in the tract. There are 12 predictor variables, mainly socio-economic.

### 2.2. Friedman #1

There are ten independent predictor variables x1,...,x10 each of which is uniformly distributed over [0, 1]. The response is given by,

$$y = 10 \sin(\pi x_1 x_2) + 20(x_3 - .5)^2 + 10x_4 + 5x_5 + \epsilon$$

where $\epsilon$ is N(0, 1). We used sample size 200.

Data Source

All data sets except Friedman #1 have been downloaded from the UCI repository. The Friedman #1 was manually generated (code attached).

Additional Files

- The code for bagging classification trees can be found in bootstrapClassification.R. The same code is used for all bootstrap classification implementation on the data sets. Code is tweaked to small extent to accommodate specific requirements of each data set.
- The code for bagging regression trees can be found in bootstrapRegression.R. The same code is used for all bootstrap regression implementation on the data sets. Code is tweaked to small extent to accommodate specific requirements of each data set.
- The code to generate Friedman #1 data set can be found in generateFriedmanDataSet.R. The function generates the Friedman #1 data set and splits it into 200 training set and 1000 test set data. The functions of above code are used to implement bootstrap regression trees.
- An excel sheet having the intermediate results

# 8. References


1. http://scikit-learn.org/stable/modules/ensemble.html
2. https://pdfs.semanticscholar.org/046b/7f6b48e4d9fcf173dea0a0802d7e87b383e1.pdf
3. https://en.wikipedia.org/wiki/Ensemble_learning
4. https://en.wikipedia.org/wiki/Statistical_ensemble_%28mathematical_physics%29
5. http://ciml.info/dl/v0_8/ciml-v0_8-ch11.pdf
6. https://en.wikipedia.org/wiki/Bootstrap_aggregating
7. https://www.stat.berkeley.edu/~breiman/bagging.pdf
8. https://stats.stackexchange.com/questions/18891/bagging-boosting-and-stacking-in-machine-learning