Ensemble Methods and Random Forests

Vaishnavi S

May 2017

1 Introduction

We have seen various analysis for classification and regression in the course. One of the common methods to reduce the generalization error is by using ensembles. We saw an example of this in the AdaBoost algorithm in one of the assignments. A way of understanding how classifiers and regressors work, is by looking at the bias-variance decomposition. After visiting that, we can view ensemble methods as trying to reduce either the bias, the variance or both. A particular ensemble method, the random forest is then introduced, and some theoretical results on the generalization error and variance are presented in the final sections.

2 Bias Variance Decomposition

The bias variance decomposition is a method of decomposing the generalization error of an algorithm. It is more natural in the context of regression than for classification. However, such decompositions have also been developed and used for classifiers, under certain assumptions. In the context of regression, the decomposition can be used to understand if the set of regressors being used are an overkill, or are insufficient for the task at hand.

2.1 Bias Variance Decomposition - Regression

In the regression problem, given a model $\hat{f}_n$, the expected generalization error at point $X = x$ is

$$\begin{aligned}
Err(\hat{f}_n(x)) &= \mathbb{E}_{Y|X=x}(y - \hat{f}_n(x))^2 \\
&= \mathbb{E}_{Y|X=x}(y - f_B(x) + f_B(x) - \hat{f}_n(x))^2 \\
&= \mathbb{E}_{Y|X=x}(y - f_B(x))^2 + \mathbb{E}_{Y|X=x}(f_B(x) - \hat{f}_n(x))^2 \\
&\quad + \mathbb{E}_{Y|X=x}(2(y - f_B(x))(f_B(x) - \hat{f}_n(x))) \\
&= \mathbb{E}_{Y|X=x}(y - f_B(x))^2 + \mathbb{E}_{Y|X=x}(f_B(x) - \hat{f}_n(x))^2 \\
&= Err(f_B(x)) + (f_B(x) - \hat{f}_n(x))^2
\end{aligned}$$

where $f_B(x)$ is the Bayes model in regression (MMSE estimator).

Here, the first term is the irreducible error of any regressor. The second term represents how different $\hat{f}_n$ is from the Bayes model. Further, taking expectation with respect to the distribution of

samples, the second term becomes

$$\begin{aligned}
&\mathbb{E}(f_B(x) - \hat{f}_n(x))^2 \\
&= \mathbb{E}(f_B(x) - \mathbb{E}\hat{f}_n(x) + \mathbb{E}\hat{f}_n(x) - \hat{f}_n(x))^2 \\
&= \mathbb{E}(f_B(x) - \mathbb{E}\hat{f}_n(x))^2 + \mathbb{E}(\mathbb{E}\hat{f}_n(x) - \hat{f}_n(x))^2 \\
&= (f_B(x) - \mathbb{E}\hat{f}_n(x))^2 + \mathbb{E}(\mathbb{E}\hat{f}_n(x) - \hat{f}_n(x))^2
\end{aligned}$$

since $\mathbb{E}(\mathbb{E}\hat{f}_n(x) - \hat{f}_n(x)) = 0$.

Through this expression, we can understand the generalization error better . The first term measures the discrepancy between the average predication and the Bayesian prediction at a fixed point x. That is, how good the expected predictor is at the point x. This is termed the bias of the predictor. The second term represents for the variability of the predictions at $X = x$ over all models. Thus, this term will be low if the variance of the estimators is low.

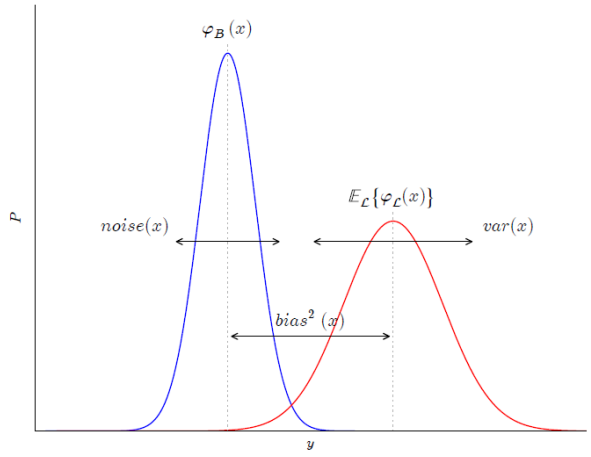

Figure 1: Irreducible error(noise), bias and variance at $X = x$, Ref [1]

Combining the above expressions we have

$$\mathbb{E}(Err(\hat{f}_n(x))) = Err(f_B(x)) + (f_B(x) - \mathbb{E}\hat{f}_n(x))^2 + \mathbb{E}(\mathbb{E}\hat{f}_n(x) - \hat{f}_n(x))^2$$

That is, the expected generalization error can be decomposed into three terms, namely, (1) the irreducible error, (2) the bias between the predictor and the Bayes optimal predictor, and (3) the variance of the estimators at the point. This is also depicted in Figure 1 above.

Therefore, to reduce generalization error we can (1)reduce bias and (2)reduce variance as defined above. It should be noted that in this framework, $X = x$ is considered fixed, and the variation with respect to noise is being recorded.

2.2 Understanding the fitting ability of regressors

The bias variance decomposition can be used in practice to figure out if the set of regressors is overfitting, or underfitting the desired function. To gain insight, consider the case of predicting a

cosine function with polynomials of varying degree. Figure 1 below shows the bias and variance for degrees 1, 5 and 15.

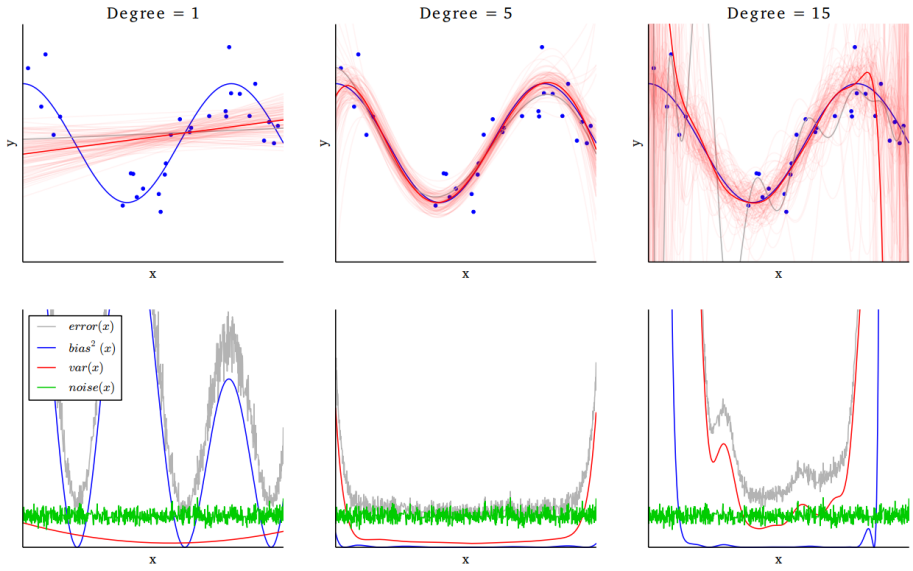

Figure 2: Bias variance decomposition of expected generalization error for polynomial of degrees 1, 5 and 15 in fitting a cosine function, Ref [1]

In the figure, the top row depicts the regressors. The blue curve is the Bayes optimal regressor, the light red ones are the regressors on different training sets, and the dark red line is the average of all those regressors. By looking at the top row alone, we can understand that the degree 1 polynomial is not enough to find the cosine function, and degree 15 polynomial will catch the tiny variations due to noise.

The bottom row depicts the bias-variance decomposition. It can be observed that there is a high bias and low variance for the degree 1 polynomial, high variance and somewhat low bias in the case of the degree 15 polynomial, while the degree 5 polynomial has both low variance and low bias. Therefore, the degree 5 case is desired. The degree 1 case is a case of underfitting, and degree 15 is overfitting.

2.3 Bias Variance Decomposition - Classification

The bias variance decomposition can also be applied in the context of classifiers, under some conditions. For example, for the 0-1 loss and binary classification (if there is some averaging process involved), the expected generalization error at $X = x$

$$\mathbb{E}(Err(\hat{f}_n(x))) = \mathbb{P}(f_B(x) \neq y) + Q\Big(\frac{0.5 - \mathbb{E}(\hat{p}_L(Y = f_B(x)))}{\sqrt{\mathbb{V}(\hat{p}_L(Y = f_B(x))}}\Big)\Big(2\mathbb{P}(f_B(x) = y) - 1\Big)$$

where $Q(x)$ is cumulative distribution of the standard normal distribution and $\hat{p}_L$ represents the conditional class probability. For more details, please refer [1].

Therefore, from the above expression, if the expected probability estimate $\mathbb{E}(\hat{p}_L(Y = f_B(x)))$ for the true class is > 0.5, a reduction in variance results in reduced total error.

3 Ensemble Methods

With the insights obtained using the bias-variance decomposition, we have an idea how to reduce the generalization error - (1)reduce bias and/or (2) reduce variance. The idea behind ensemble methods is to achieve this by

- Training multiple learners to solve the same problem
- Introducing random perturbations into learning procedure to produce several different models for same training set
- Combining predictions of these models to form prediction of ensemble

3.1 Examples of Ensemble Methods

Popular methods of ensembles include

- Boosting: Combining multiple weak learners and boosting them based on their cumulative performance (e.g. AdaBoost which adaptively boosts learners). These methods aim to reduce bias and thus achieve reduced generalization error
- Bagging/Bootstrap Aggregating: Combining multiple learners with the aim of reducing the variance and leaving bias unaffected. These provide training data to the learners by picking uniformly at random, with replacement, from the set of all training samples. (e.g. Random Forests introduced by Breiman)

3.2 Why do ensembles work?

Before introducing random forests and decision trees, we look at possible explanations as to why ensembles work. As presented by Thomas Dietterich in [2], there are three possible reasons[Figure 3]:

- Statistical: It is possible that the learning algorithm can find many hypothesis that all give same accuracy on training data. Combining these hypotheses might do better.
- Computational: Many algorithms can get stuck at local optima. Using many different starting points helps getting stuck at the same local optima. Therefore multiple learners could help/
- Represenational: The true function may not be in the chosen hypothesis space. Weighted sums expand the space of representable functions, increasing the chance of including the true function in the search space.

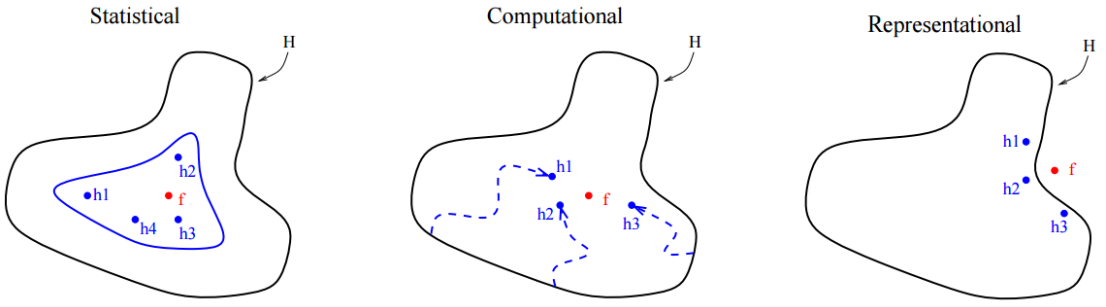

Figure 3: Why ensembles work

4 Random Forests

So far we have seen the intuition behind ensemble methods. Now we present the constituents of random forests (decision trees), how the random forests are constructed, and some theoretical results.

4.1 Decision Trees

Decision trees are tree based classifiers/regressors. They arise naturally in the context of classification.

To build one:

- Start at root node corresponding to all samples
- Split current node based on variable that maximizes a split criterion
- Split all such nodes at current level
- Repeat at children nodes
- Continue till stopping criterion is true

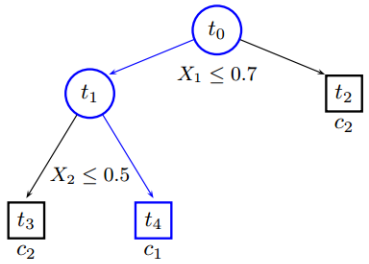

Figure 4: A decision tree

Commonly used split criteria include maximum information gain, Gini impurity

It is to be noted that decision trees have a high chance of overfitting. That is, they have a high variance, low bias. Hence, they form good candidates for ensembles which aim to reduce variance while not affected bias by much.

5 Random Forests

Random forests are learners which are a collection of decision trees. Breiman introduced bagging and random forests in the early 2000s. Ever since random forests have become a popular learning technique. They are particularly useful because not only do they have a good performance in practice, one can pinpoint variables and variable values corresponding to nodes and therefore make sense of what is going on. This is unlike the case of neural networks where the learner is, very often, treated like a magical black box.

5.1 Bagging

The idea of bagging is to generate training sets for the individual learners of the ensemble. As an example, consider random forests to be a collection of (say M) decision trees. Bagging is used to generate training set for the decision trees as follows.

- Choose N cases by drawing uniformly at random with replacement from the training set of samples $(Z_1 \dots Z_N)$
- Repeat for each tree $T_i, i = 1 \dots M$

Further constraints are imposed on the constituent decision trees, to split only along a subset of all possible variables. Therefore, these fall under the category of subspace sampling methods.
After the trees are constructed, to evaluate the output of the random forest, aggregate results of all decision trees e.g. The random forest regressor averages the output of all decision trees, thus $f_{RF}(x) = \frac{1}{M}\sum_{m=1}^{M} f(x, \Theta_m)$ where Θ_m parametrizes the m^{th} tree.

5.2 An upper bound on generalization error

In the case of regression, an upper bound on the generalization error can be obtained in terms of the individual trees and the amount of correlation between them. While this bound is loose, it gives an idea of how to construct random forests for low generalization error.

For a regression problem, the mean-squared generalization error for predictor f(x) is given by $\mathbb{E}_Z(y - f(x))^2$. Thus, for random forests, the error is given by

$$\mathbb{E}_Z(y - f_{RF}(x))^2 = \mathbb{E}_Z(y - \frac{1}{M}\sum_{m=1}^{M} f(x, \Theta_m))^2 \xrightarrow[M\to\infty]{a.s.} \mathbb{E}_Z(y - \mathbb{E}_\Theta f(x, \Theta))^2$$

The last term in the above expression is defined to be the optimal generalization error of random forests. That is, $L^*_{RF} = \mathbb{E}_Z(y - \mathbb{E}_\Theta f(x, \Theta))^2$. The optimal generalization error of tree is given by $L^*_{tree} = \mathbb{E}_\Theta \mathbb{E}_Z(y - f(x, \Theta))^2$. The following theorem relates these two quantities.

Theorem 5.1 ([3] Breiman, 2001) *Let* $L^*_{RF} = \mathbb{E}_Z(y - \mathbb{E}_\Theta f(x, \Theta))^2$ *and* $L^*_{tree} = \mathbb{E}_\Theta \mathbb{E}_Z(y - f(x, \Theta))^2$. *Assume that* $\forall\Theta,\ \mathbb{E}_Z(y - f(x, \Theta)) = 0$. *Then*

$$L^*_{RF} \leq \bar{\rho} L^*_{tree}$$

where $\bar{\rho}$ *is the weighted correlation between residuals defined below and* Θ, Θ' *are independent.*

Proof:

$$\begin{aligned}
L^*_{RF} &= \mathbb{E}_Z(y - \mathbb{E}_\Theta f(x, \Theta))^2 \\
&= \mathbb{E}_Z(\mathbb{E}_\Theta(y - f(x, \Theta)))^2 \\
&= \mathbb{E}_\Theta \mathbb{E}_{\Theta'} \mathbb{E}_Z(y - f(x, \Theta))(y - f(x, \Theta')) \\
&\overset{(a)}{=} \mathbb{E}_\Theta \mathbb{E}_{\Theta'}(\rho(\Theta, \Theta') sd(\Theta) sd(\Theta')) \\
&\overset{(b)}{=} \bar{\rho}(\mathbb{E}_\Theta sd(\Theta))^2 \\
&\overset{(c)}{\leq} \bar{\rho} \mathbb{E}_\Theta \mathbb{E}_Z(y - f(x, \Theta))^2 \\
&= \bar{\rho} L^*_{tree}
\end{aligned}$$

where $\rho(\Theta, \Theta') = \dfrac{\mathbb{E}_Z((y - f(x, \Theta))(y - f(x, \Theta')))}{sd(\Theta) sd(\Theta')}$, $\bar{\rho} = \dfrac{\mathbb{E}_\Theta \mathbb{E}_{\Theta'} \rho(\Theta, \Theta')}{\mathbb{E}_\Theta(\sqrt{\mathbb{E}_Z(y - f(x, \Theta))^2})^2}$ and $sd(\Theta) = \sqrt{\mathbb{E}_Z(y - f(x, \Theta))^2}$

The correlation coefficient $\rho(\Theta, \Theta')$ defines the correlation between the residuals $y - f(x, \Theta)$ and $y - f(x, \Theta')$. The weighted correlation coefficient $\bar{\rho}$ weights the correlation $\rho(\Theta, \Theta')$ further.

(a) follows by introducing the correlation coefficient. (b) follows by introducing the weighted correlation coefficient. (c) follows from the non-negativity of variance.

The above gives the idea that as long as the correlation between the error of two different trees is low, it is alright for the individual decision trees overfit. Therefore, the randomization in the construction of the random forest should ensure a low correlation.

5.3 Variance of random forests estimates (Regression)

Does the variance of the estimates of random forests actually go to zero, as desired?

The following theorem is presented as part of the proof of that the generalization error goes to zero. In particular, the authors employ the bias-variance decomposition and show that both terms go to zero. However the rate of decay is poor, and the result is presented in a very restricted context.

Theorem 5.2 *[Proposition 2, [4] Gerard Biau, 2012] Assume that X is uniformly distributed on $[0,1]^d$, sampling without replacement, and some other constraints*

$$\mathbb{E}(\mathbb{E}\hat{f}_n(x) - \hat{f}_n(x))^2 = O\Big(\Big(\frac{1}{n}\Big)\Big(\frac{k_n}{\log(k_n)^{S/2d}}\Big)\Big)$$

where k_n represents the number of leaves, and S is the sparsity of the function to be estimated

In the extreme case where $k_n = n$ and $S = d$, the variance term $= O\Big(\frac{1}{\sqrt{\log n}}\Big)$ While this indicates a very slow rate of convergence, there is often sparsity in the representation of the function to be estimated. Therefore, $S < d/2$, which yields better rates.

According to the authors, the proof reveals that the log term is a by-product of the Θ-averaging process, which appears by taking into consideration the correlation between trees.

6 Simulations

As part of learning about random forests, some simulations are presented here.

6.1 Bias-Variance Decomposition

Figure 5 depicts the simulation of regression using trees(left) and bagged trees(right). There is significantly lesser peaks in the red and green curves corresponding to error and variance respectively, for the case of bagged trees.

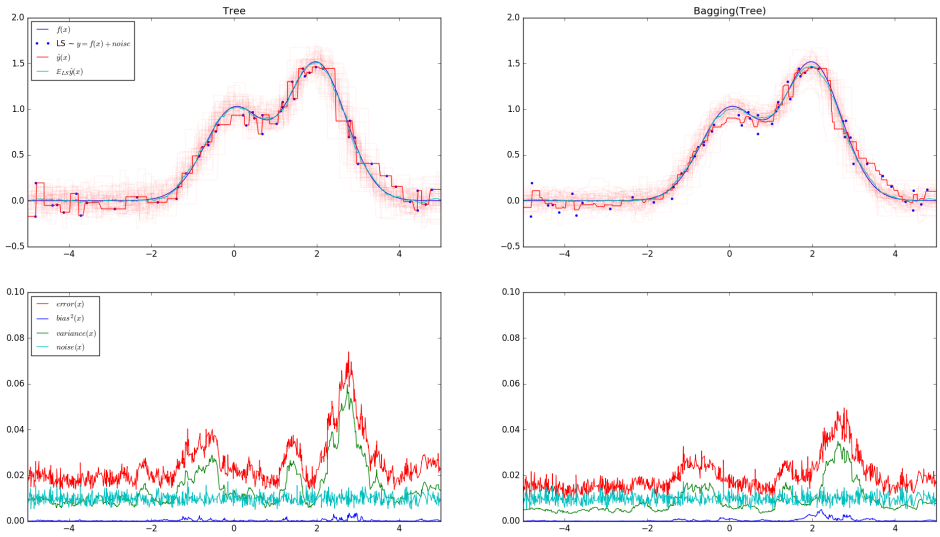

Figure 5: Bias-variance decomposition of trees without and with bagging

6.2 Classifiers on the Iris dataset

We have worked with the Iris dataset on several of our Python problems. Here, the classification of the Iris dataset, based on subset of features is presented for decision trees, random forests, and Adaboost algorithms.

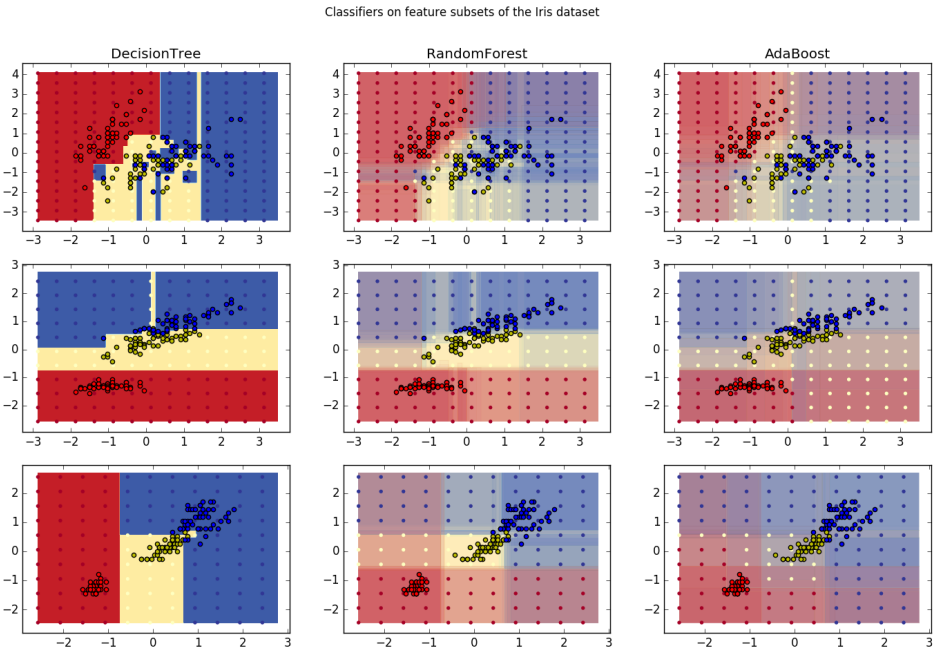

Figure 6: Comparing Decision Trees, Random Forests and AdaBoost

It is observed that decision trees have sharp boundaries, often overfitting in multiple places. Random forests have a smoother transition between two classes. Adaboost appears to perform almost as well as random forests on this dataset.

6.3 Out of bag estimate of error

In the case on online learning algorithms we saw that the error on a new point yields an idea about the generalization error of the learner. The same is true in the case of random forests. Here, we can test the trained random forest on samples not allotted to the decision tree, which yields the out-of-bag estimate of the error. In Figure 7, the out-of-bag error estimates are compared for random forests constructed with different number of features/variables for splitting along during each split of the tree. The comparison is with respect to the number of estimators used. Looking at such a plot helps in picking the number of trees you would want to include in the random forest.

It is interesting to note that the OOB error when the max-features is none, that is, when all features are used, is higher than in the other two scenarios.

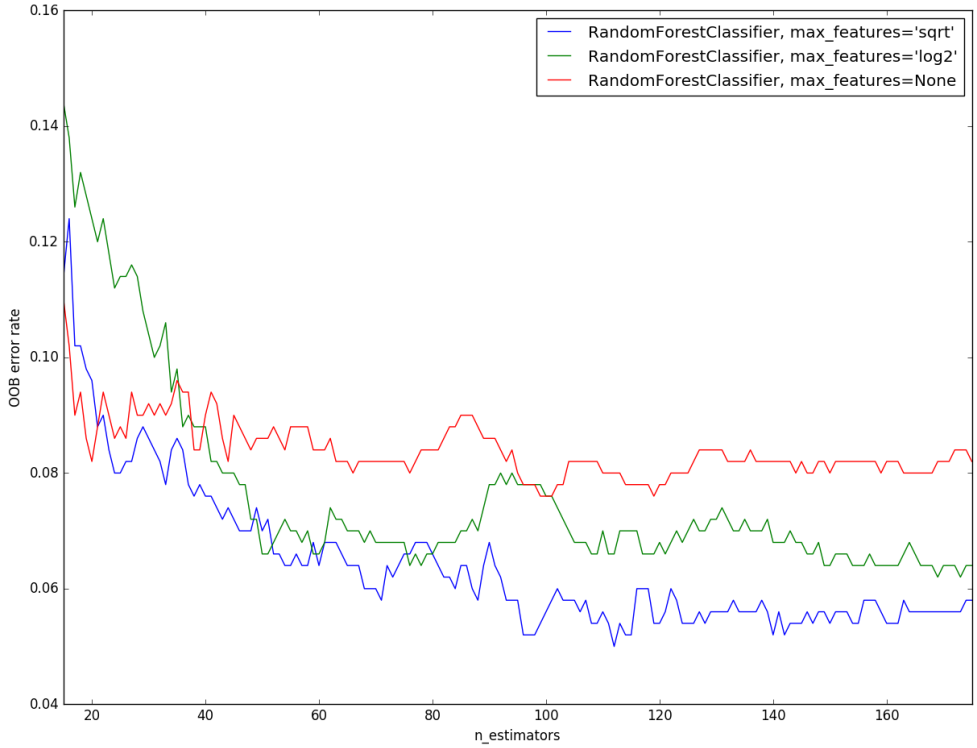

Figure 7: Out of Bag error estimate

7 Conclusions

In this project one of the popular learning algorithms is studied. The generalization error bounds for certain cases are presented. While the framework is similar to the one seen in class, the proof methodology is somewhat different. The proofs rely on the bias-variance decomposition presented, as well as the results on decision trees and ensemble methods existing in literature. The claim that the random forest should reduce the variance of the learner is also presented. This turns out to be true in a certain regression case. In other contexts, the performance of random forests is not proved sufficiently. Some simulations are presented.

References

[1] Gilles Louppe. Understanding random forests: From theory to practice, chapters 3,4. *arXiv preprint arXiv:1407.7502*, 2014.

[2] Thomas G Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2000.

[3] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[4] Gerard Biau. Analysis of a random forests model. *Journal of Machine Learning Research*, 13(Apr):1063–1095, 2012.