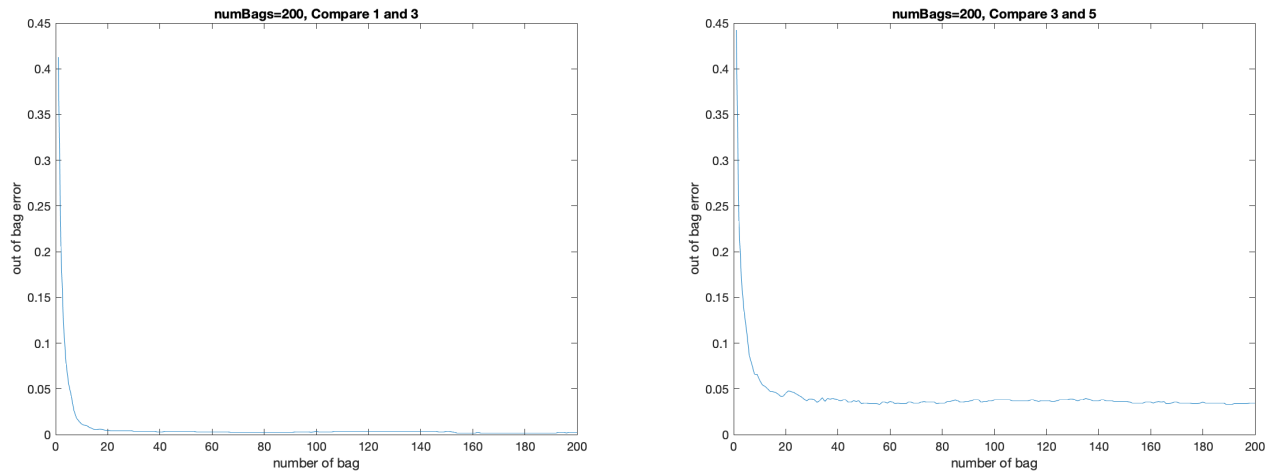


Problems

1. BaggedTrees Implementation

(a)



The above graphs are the results that using random forest method to learn a tree for each bag and calculate the out-of-bag error in the group of 1 and 3 digits and 3 and 5 digits. As we can see, the out-of-bag error decreases with the number of bags increases.

(b)

As we run the OneThreeFive file, the report is as follows,

```
Working on the one-vs-three problem...
```

```
The cross-validation error of decision trees is 0.0084
The OOB error of 200 bagged decision trees is 0.0024
```

```
Now working on the three-vs-five problem...
```

```
The cross-validation error of decision trees is 0.0634
The OOB error of 200 bagged decision trees is 0.0346
^^
```

Table 1: Results table of 1 v.s. 3 and 3 v.s. 5

	1 v.s. 3	3 v.s. 5
Cross-Validation	0.0084	0.0634
Out-of-bag Error	0.0024	0.0346

(c)

As we learn one decision tree from our training dataset, and use test dataset to estimate test error, the result is as follows,

```
Working on the one-vs-three problem...
```

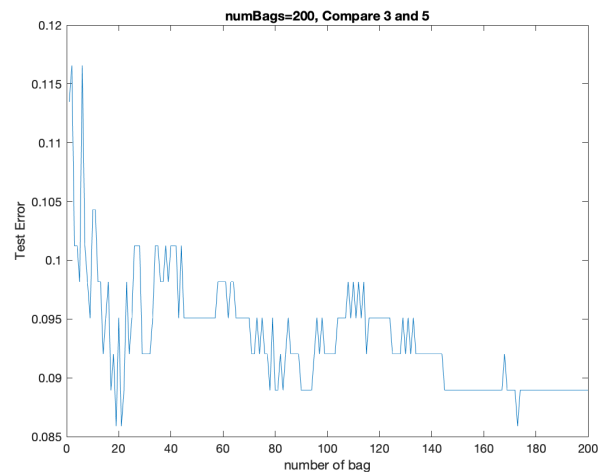
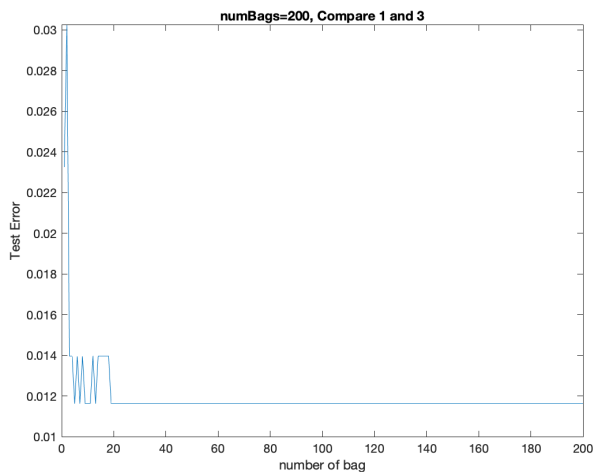
```
The test error of one decision trees is 0.0163
The test error of 200 bagged decision trees is 0.0116
^^
```

```
Working on the Three-vs-Five problem...
```

```
The test error of one decision trees is 0.1258
The test error of 200 bagged decision trees is 0.0890
^^
```

Table 2: Test Error of 1 v.s. 3 and 3 v.s. 5

	1 v.s. 3	3 v.s. 5
One Tree	0.0163	0.1258
Ensemble of 200 Trees	0.0116	0.0890



(d)

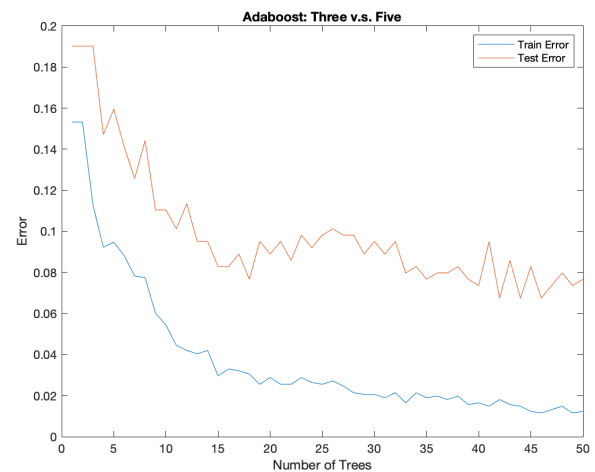
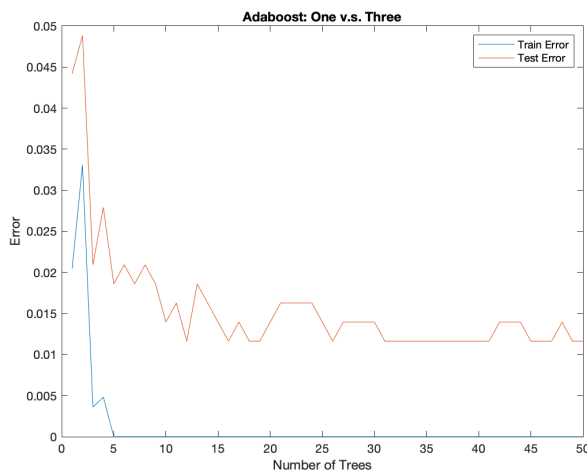
For part (b), the result we got from cross-validation and bagging are different. It's because, bagging averages models in order to reduce the variance, while the cross-validation evaluate a number of surrogate models assuming they are all equivalent (good surrogate) for the actual model. The reason that cross-validation underperforms compared to bagging is because in a decision tree, using cross-validation could get a totally different tree each time we resample the training data, that means the variance is pretty high. And bagging is using multiple models to make more accurate predictions than any individual model.

For part (c), which is an example that reinforce the point above, when we only have one decision tree, the test error can be enormously large. While we increased the number of bags, that could reduce the test error.

In both out-of-bag error and test error, there's no significant to overfitting.

Problems

2. Adaboost



Adaboost is an algorithm that's to combine multiple weak learner into a stronger one. Each tree only uses one feature to do the split, that we call a stump, and there are a bunch of stumps and each stump has different importance and each time we split, we're going to focus more on the misclassified data points by increasing the weight on misclassified data points to let the next stump to adjust according to the sample rate. Basically, we use Gini index to decide which feature we're going to use to split our data at each depth, and by updating the sample weight, we can calculate weighted Gini Index to determine which variable is going to be used to split the next stump. The weighted Gini Index would put more emphasis on correctly classifying the samples that are misclassified by the previous stump since the samples have larger weight. The alternative is we can use sample weights to create a new dataset that reflects those weight.

After a briefly intro to the Adaboost, we can explain the result in a more intuitive way.

What we can observe is that in both cases, **One v.s. Three** and **Three v.s. Five**, the training error is lower than test error. This is pretty intuitive, after all, we use our training dataset to build our decision tree so we can expect the training error would be lower than the error that comes from by using a different dataset, that is, test dataset.

The other fact is, by increasing the number of trees we can observe the error decreases. It should not be surprise as a tree with more depth, the higher chance that our tree could correctly classify our data point. And by using Adaboost, since we're focusing on the misclassified data points, there's a chance that we may suffer from overfitting since the algorithm will put more weight on the misclassified data points and that would affect how we grow our tree.

Collaboration Statement

I collaborate this assignment with Jingru Hu, 466024, jingruhu@wustl.edu.