Assignment 2

Mukul Sati [msati3@gatech.edu]

March 1, 2016

1 Random forests trained with bagging

I carried out two iterations of the following experiment: I trained multiple random forests, with varying number of decision trees (I used the implementation in sklearn). I let the decision trees grow unconstrained during the first iteration, and limited their depth to 5 levels during the second. I create bagged sets of size 1/3 of the training set for both the iterations.

1.1 Wine dataset

For the wine dataset, I selected a random subset of size 75% of the original data for training. I did not ensure that the samples in the training set are equally distributed for each label. While this is sub-optimal as mentioned by the instructor on Piazza, I think the split of 33%, 40%, 27% amongst the classes is not substantial to adversely affect the results.

The plots for the errors for the two iterations are shown in Fig. 1.1.

The confusion matrix for the wine-data for a Random Forest with 100 depth-not-limited trees and one with 150 depth-limited (to 5 levels) trees respectively is shown in Table 1

1.2 MNIST dataset

For MNIST, I used PCA to reduce dimensionality, retaining enough principal component vectors that explain 80% of the variance in the data. I feel this gives

Label	1	2	3
1	10	1	0
2	0	21	0
3	0	0	12

Label	1	2	3
1	10	1	0
2	1	20	0
3	0	0	12

Table 1: The confusion matrix for a Random Forest with 100 depth-not-limited trees (left). The confusion for a Random Forest with 150 depth-limited to 5 trees (right). These results are for the wine-dataset.

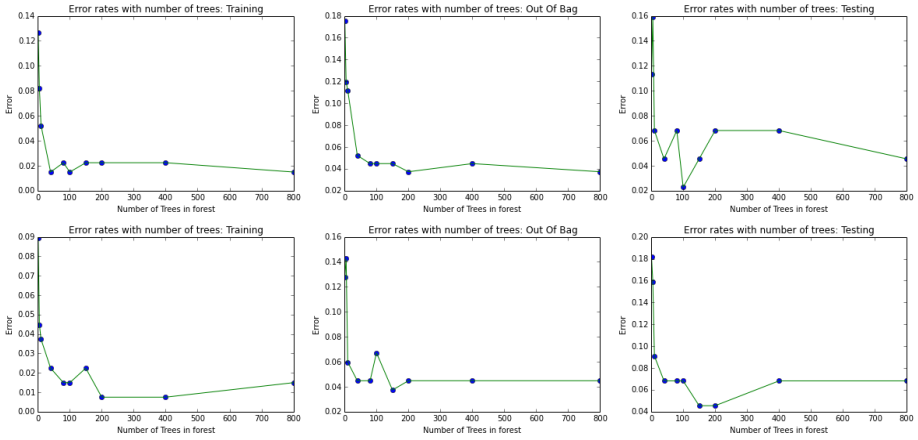

Figure 1: The errors on the (a) Training, (b) Out of bag and (c) Testing data when using unconstrained depth binary trees (top) and when using binary trees that are only allowed to grow to depth 5 (bottom) for the wine dataset

Digit	0	1	3	5
0	964	0	7	9
1	0	1126	5	4
3	4	4	972	30
5	18	1	28	845

Digit	0	1	3	5
0	852	0	26	102
1	0	1092	14	29
3	13	8	847	142
5	42	4	80	766

Table 2: The confusion matrix for a Random Forest with 150 depth-not-limited trees (left). The confusion for a Random Forest with 40 depth-limited to 5 trees (right). These results are for the MNIST dataset.

me a good balance between a performant algorithm and execution time. The plots for the errors for the two iterations are shown in Fig. 1.2.

The confusion matrix for the MNIST dataset for a Random Forest with 100 depth-not-limited trees and one with 150 depth-limited (to 5 levels) trees respectively is shown in Table 2

1.3 Discussion

My observations are in line with [1]. The training error soon plateaus in both the cases (depth limited and unconstrained decision trees). However, the out of bag error still decreases for a while post this. This may be explained in terms of margin function (the difference between the number of correct class predictions, and the maximum over the number of incorrect predictions for a given incorrect class), is driven to its limiting value of the probabilities of the two classes under consideration over the feature space. Thus, the generalization

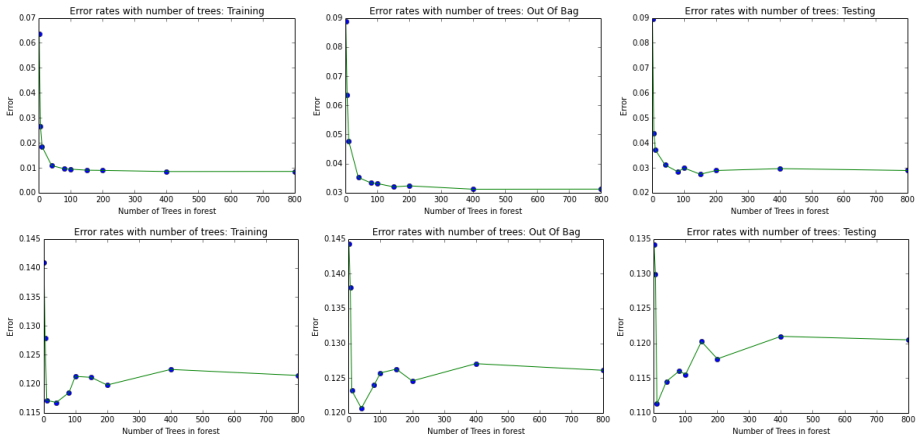

Figure 2: The errors on the (a) Training, (b) Out of bag and (c) Testing data when using unconstrained depth binary trees (top) and when using binary trees that are only allowed to grow to depth 5 (bottom) for the MNIST dataset

error may still decrease even after the training error plateaus, due to improving probability margins between confusable classes. This effect is observed in the training errors, where they decrease (see particularly, Fig. 1.2) with increasing number of trees, due to improving generalization.

2 Boosting

I have (tried to) implemented the AdaBoost.M2 algorithm as described in [2]. Using both a decision stump, and a decision tree limited to grow to depth 10 as the weak learners, I train boosted ensembles with a varying number of weak learners in them.

2.1 MNIST dataset

For MNIST, similar to bagging, I used PCA to reduce dimensionality, retaining enough principal component vectors that explain 90% of the variance in the data. I feel this gives me a good balance between a performant algorithm and execution time. The plots for the errors for the two iterations are shown in Fig. 2.1.

The confusion matrix for the MNIST dataset for an ensemble of 100 boosted decision tree stumps and one with 150 depth-limited (to 10 levels) decision trees respectively is shown in Table 3

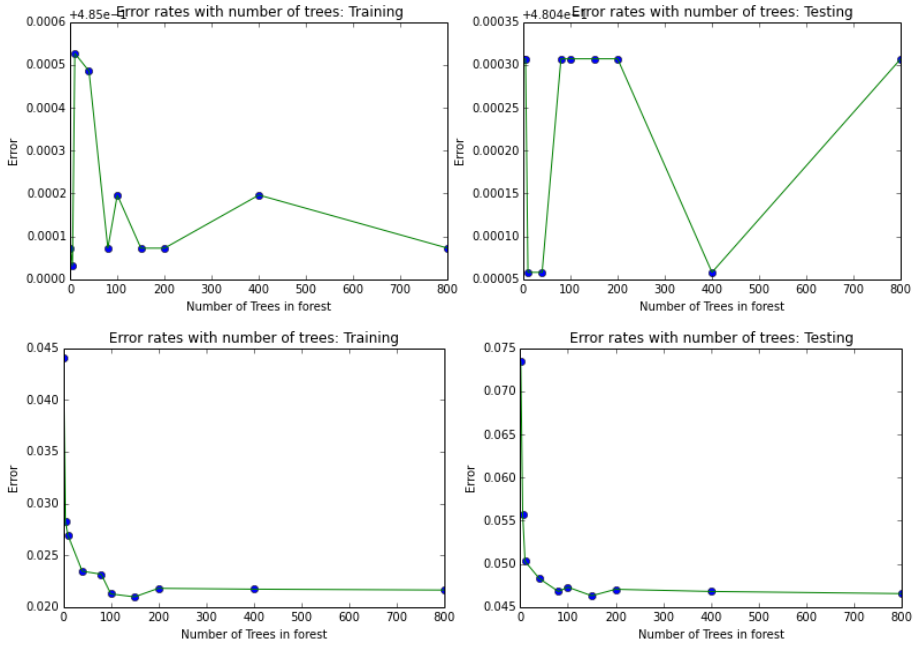

Figure 3: The errors on the (a) Training and (b) Testing data when using boosted decision stumps (top) and when using boosted decision trees that are only allowed to grow to depth 10 (bottom) for the MNIST dataset

Digit	0	1	3	5
0	0	0	980	0
1	0	1093	42	0
3	0	17	993	0
5	0	29	863	0

Digit	0	1	3	5
0	951	0	7	22
1	0	1120	9	6
3	6	7	938	59
5	22	2	46	822

Table 3: The confusion matrix for a boosted ensemble of decision tree stumps trained over 100 iterations (left). The confusion for a boosted ensemble of 150 depth-limited to 10 decision trees (right). These results are for the MNIST dataset.

2.2 Office Dataset

Not worked on

2.3 Discussion

Boosting performance using decision tree stumps is poor for the 4 class problem, as only two possible classes can be identified by each decision tree. However, this performance is better than the weak learner we started with. As expected, the performance when using decision trees with a maximum depth of 10 is much improved due to it being a stronger learner. In [3], in an analogous manner to [1], the authors attempt to explain the paradox of not-overfitting with increasing number of learners in the ensemble — contrary to what would be expected using VC-dimension based arguments — with margin theory, even drawing some parallels with SVMs. I don't observe the same results on my experiments, and I'd hazard a guess that this is due to a bug in my code.

References

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

[2] Yoav Freund, Robert E Schapire, et al. Experiments with a new boosting algorithm. 1996.

[3] Yoav Freund and Robert Schapire. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612, 1999.