Boosting Analysis

Paul Laliberte' | CSCI-5622

Problem 1. Use your AdaBoost code with Sklearn's DecisionTreeClassifier as the base learner to distinguish 4's from 9's.

All simulations to follow (and in problem 2 and 3) used a limit of 4000 data points with 400 learners, unless otherwise specified. We present a classification report and a confusion matrix for a decision stump of 1 (deemed to be the best depthed classifier as discussed below):

Figure 1: Metrics: Decision Stump of Depth-1

	precision	recall	f1-score	support
4	0.97	0.96	0.96	982
9	0.96	0.97	0.97	1009
avg / total	0.96	0.96	0.96	1991

942	40
30	979

Problem 2. Run several hundred boosting iterations with trees of depths 1, 2, and 3 (go deeper if you like) as the weak learner. Make plots of training and test error per boosting iteration, compare depths, overfitting (100% accuracy), and further iterations.

The following plots consist of decision stumps of depths 1, 2, and 3:

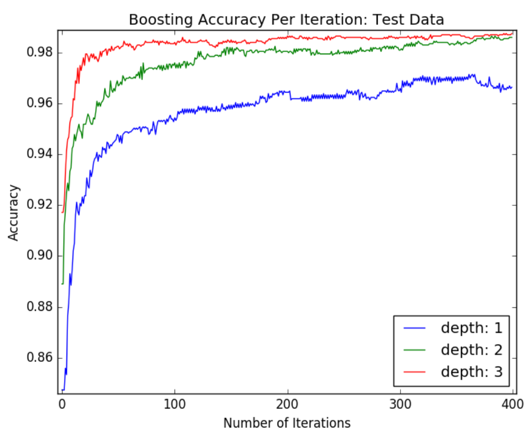

Figure 2

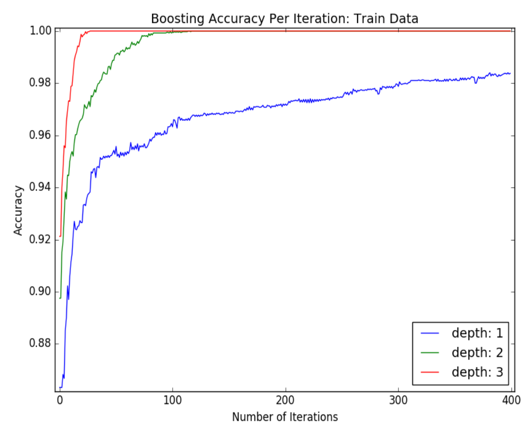

Figure 3

First notice, to no surprise, that the test data (Figure 2) gives a more reasonable outlook on what an expected prediction accuracy is to be on a hold-out test set.

This brings us to overfitting. The train set (Figure 3) is severely overfitted for depths 2 and 3, and even achieves 100% accuracy on both respective decision stumps. Surprisingly, a decision stump of 1 on the train set performs only approximately 2% better than on the test set. Even though the test set seems to have a much steadier increase in accuracy (and thus less of a chance of overfitting) than on the train set, overfitting stills seems to be prevalent in depths 2 and 3, which almost achieve 100% accuracy. For depths 2 and 3 an acceptable stopping point to minimize the chance of overfitting is close to 150 iterations, whereas for a depth of 1 it would be around 320. Since a decision stump of depth 1 seems to provide a very respected accuracy and the lowest chance of overfitting, we can assume that, on this data set, that a depth of 1 would be ideal to use.

Problem 3. Try another classifier from Sklearn as the weak learner (e.g. a Perceptron) and repeat the steps above. How does your choice of weak learner compare to decision trees?

We ran our AdaBoost algorithm on a perceptron classifier, i.e. one perceptron and not a neural net, with varying parameters (default, l2-regularization, and epoch variations). The default perceptrons were run on only 1000 data points instead of 4000, and resulted in very sporadic accuracies (Figure 4 and 5). The perceptron run on the training data did achieve an accuracy of 100%, while the perceptron on the test data actually decreased in accuracy. However, if we add the parameters of l2-regularization and varying epochs (and 4000 data points) with the perceptron we get a much better outlook (Figure 6). As with the decision stumps, the perceptron with parameters may be showing some signs of overfitting (epochs 30 and 50), but it is hard to say as they do not reach a level of 100% accuracy like the decision stumps and the default perceptron. A perceptron with l2-regularization and 50 epochs looks very similar to a decision stump of depth 1, and thus these parameters seem to make a perceptron just as capable as a decision stump on this data set. The perceptron with an epoch of 10 increases too slow (if at all) and the perceptron with epoch 30 increases too fast, but does correct for this in later iterations.

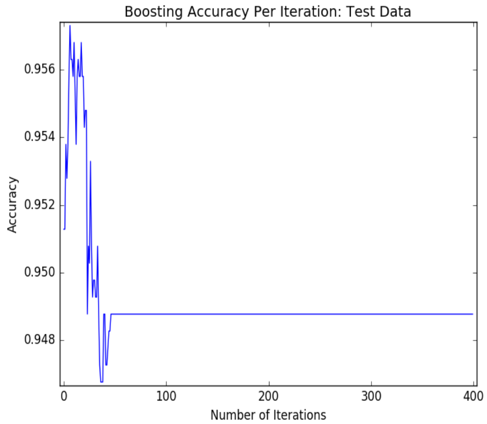

Figure 4: Perceptron: default settings, 1000 data points.

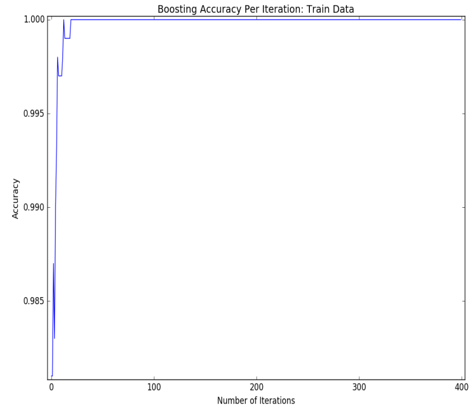

Figure 5: Perceptron: default settings, 1000 data points.

Figure 6: Perceptron: l2 regularization, varying epochs, 4000 data points.