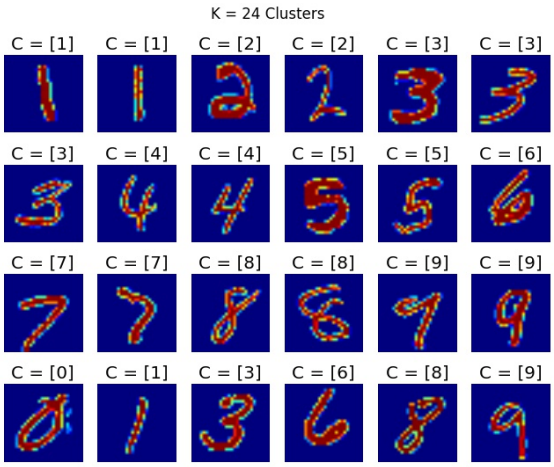
HW6: Random Forests & Boosting Weak Learners

In this exercise we will explore Random Forests and boosting weak learners using AdaBoost. By now you should understand how to find the optimal regularization parameters of a learning algorithm so I will be providing less instruction expecting you to take your first solo flight!

We will explore that same datasets as we did before, hand-written digits and newsgroup text. The MNIST dataset is comprised of 28x28 pixel images of hand-written digits from the U.S. Census Bureau. Each instance is made up of 784 features (28 x 28 pixel images), each representing a pixel color value between 0 and 1. There are 10 classes to classify, digits 0-9. Here is an example of the hand-written digits you will classify.



The 20 newsgroup dataset is comprised of class 0 (“comp.graphics”) and class 1 (“comp.windows.x”) new wire segments. Some examples are:

y=0 articl repli program work comput distribut system mail code softwar applic inform call internet fax point includ group number address type center id research current video usa develop engin david robert page design accept author contact gmt

y=0 nntp write program file graphic distribut time code find softwar call thing point d sourc includ give function ftp librari inc draw suggest lot wrote site implement stuff access handl object book place year fast output exist routin document robert limit ad pretti easi mac level

y=1 nntp nt articl program find ve make internet server point motif interest gener manag start inc type widget client lot id expolcsmitedu case put result memori xpert handl event back place termin long network design perform great

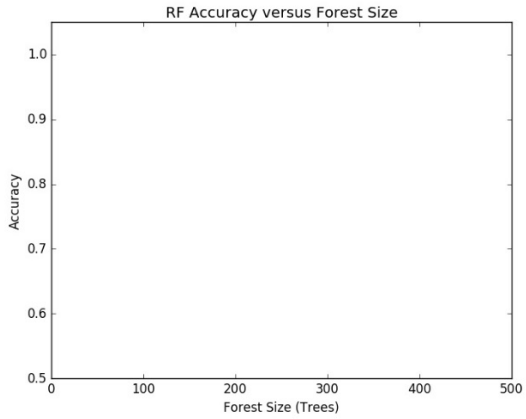
y=1 nntp write nt window distribut mail softwar ca motif unix manag usa engin david handl object ms design xt

The features are 2997 distinct words (bag of words model) where each news wire case’s feature space would be the words that are found in case. (word present = 1, word absent = 0).

HW6: Random Forests & Boosting Weak Learners

1. Random Forest

The first learner that you will implement will be a Random Forest. A Random Forest has several regularization hyper-parameters to explore, namely, forest size, purity measure, number of random split features, and tree depth. Your implementation should evaluate your training error, and test error using 5-fold cross validation as you have done in the past. You will then graph your results on individual graphs for both the MNIST and 20NG datasets. Your graphs should compare forest size to accuracy as in the shell below. It should have a legend and show the training and testing error of the models you develop varying your hyper-parameters. The objective is to find the right combination of hyper-parameters that produces the strongest Random Forest learner. You will need to determine what those are on your own. To give you a general sense, I evaluated 36 – 42 models represented by 6 or 7 paired lines (12 – 14 training and test lines) on my graph.



(a) Record the test and training accuracy for each hyper-parameter setting for each dataset.

Hyper-Parameter Value	Training Accuracy	Test Accuracy

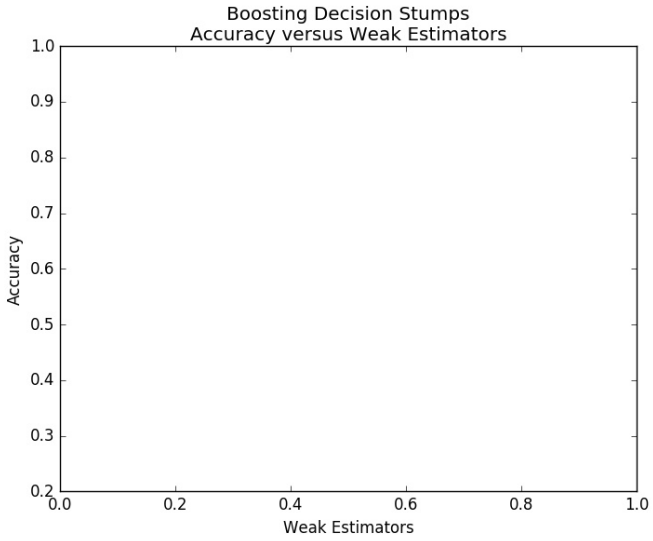
(b) Using your two graphs, determine the optimal regularization parameter value for each dataset. Explain why you think it is the optimal value?

(c) Again, looking at the graphs determine if you see evidence of overfitting or underfitting? If so, for which regularization value(s)? Explain if these results make sense or if they seem odd.

HW6: Random Forests & Boosting Weak Learners

2. AdaBoost

Now you will boost Decision Stumps as a weak learner. To produce a Decision Stump you will set the tree depth to a maximum of 1. This should be constant and not vary. Here you have several regularization hyper-parameters to consider, namely, number of weak learners, and learning rate. Your implementation should evaluate your training error, and test error using 5-fold cross validation as you have done in the past. You will then graph your results on individual graphs for both the MNIST and 20NG datasets. Your graphs should compare weak estimators to accuracy as in the shell below. It should have a legend and show the training and testing error of the models you develop varying your hyper-parameters. Again, the objective is to find the right combination of hyper-parameters that produces the strongest set of boosted weak learners. You will need to determine what those are on your own. To give you a general sense, I evaluated 24 – 30 models represented by 4 or 5 paired lines (8 – 10 training and test lines) on my graph.



(a) Record the test and training accuracy in a table like the example table for the Random Forest. Produce this for each of the two datasets.

(b) Using your two graphs, determine the optimal model and its optimal regularization parameter values for each dataset. This means you should select 1 best model for each dataset and explain why you think it is the optimal model?

(c) Again, looking at the graphs for each dataset, determine if you see evidence of overfitting or underfitting? If so, for which regularization value(s) in each model? Explain if these results make sense or if they seem odd.

HW6: Random Forests & Boosting Weak Learners

3. Further Analysis of Results

(a) Which overall model is the best across the Random Forest and boosted Decision Stumps? Explain why you selected the model that you did beyond just looking at the accuracy.

What to Hand in

You should hand in a *.pdf containing all of the necessary write-up items along with your evaluation results and graphs. Also hand in your code. Zip this all up.