Homework 2 | Ensemble Learning

Course: CS7616
Due: Sun, Feb 28, 11:55pm

Random Forests

In this problem set you will be implementing **Random Forest** (RF) classifiers and comparing their performance with different tree sizes and number of trees. RFs were discussed in the lectures and the algorithm can be found in the slides. In the original formulation each of classification tree is grown to its full-extent. You are going to implement a variant where every tree can be grown to maximum depth of five. This will be in addition to implementing the original. You can use a library such as sklearn for only just the decision tree, or for extra credit you may create your own decision tree using a metric like shannon entropy or the gini index.

For each of **Wine** and **MNSIT** datasets, you are required to do the following:

1. Split the data into 75% training and 25% test sets.
2. Train two types of RFs, one in which each of the decision trees is grown to the full extent, and one in which the maximum tree depth is five. Do this for number of trees (B) from 1 to a reasonable number (minimum five hundred). You can choose the reasonable number by investigating the B at which the errors reach a plateau or starts increasing. [30 points]
3. Show the final confusion matrix for both types of RF described above. [5 points]
4. For each of the two types plot the training error, test error, and out-of-bag error w.r.t. the number of trees. [5 points]
5. Comment on the trends you observed with increasing B for the two types of RFs. [5 points]
6. Analyze and explain the trends you noted. [5 points]

You might like to train the RF with the maximum number of trees first and then sample from it to simulate RFs with less trees. We did not discuss the parameters m (the number of randomly chosen dimensions at each split) here, nor the size of each of the bootstrap sets. You are expected to make your own choice and reason for them

> **Out-Of-Bag Error:** Let the training set be T and the bootstrap training sets T_k for the k^{th} classifier. For each data point (x, y) in T aggregate over the votes of only those classifiers that were not trained on (x, y). This is the out-of-bag classifier. The out-of-bag error is then the error of this classifier averaged over the training set.

Boosting

Another committee method is Boosting where each member is trained on the same training set. However, the data points are weighed differently and each of them is only a weak learner. For this part you will implement a multiclass variant of AdaBoost, such as **AdaBoost.M2**,**Sequential AdaBoost**, etc. The algorithm for these are given in the slides and the papers we have discussed. We will again compare two different types of committee members.

For each of the **MNIST** and **Office** dataset, you are required to do the following:

1. Split the dataset into 75% and 25% of training and test sets.
2. Implement AdaBoost with two types of weak learners, one in which the learner is a decision stump (one level tree), and the other one in which the maximum tree depth is ten. Do this for number for a reasonable number of iterations. You can choose the reasonable number by investigating when the test errors start increasing. [30 points]

3. Show the final confusion matrix for both types of boosting described above. [5 points]
4. For each of the two types plot the training error, test error, w.r.t. the number of iterations. [5 points]
5. Comment on the trends you observed with increasing iterations for the two types of learners. [5 points]
6. Analyze and explain the trends you noted. [5 points]

Datasets

This are just the source links for technical reference. Please go to the **T-square resources** page to download the pre-processed data ready for you to use in this assessment.

Dataset	Description	Data
Wine Data Set	[Description]	[Data]
MNIST	[Description]	[Data]
Office Dataset	[Description]	[Data]

Submit

Submit a your code and a PDF (or working Jupyter Notebook) report with results in a zip file. You can use any programming language, but we will prefer `Python` or `Matlab`. The code should be easily runnable. Please include a README.md file that describes how the TAs are to run your code. Note, you do not need to include the original data sets in the zip file (the is also a Tsquare upload limit the prevents this), however please make it easy for the TAs to run you code, such that pasting the extracted Tsquare dataset folders into the root project zip folder would work, e.g something like this:

```
$ unzip Doe-John-HW2.zip
$ unzip wine.zip -d Doe-John-HW2/wine
$ unzip MNIST.zip -d Doe-John-HW2/MNIST
$ unzip office.zip -d Doe-John-HW2/office
$ tree Doe-John-HW2
Doe-John-HW2
|-- HW2.ipynb
|-- HW2.pdf
|-- MNIST
|   |-- test.csv
|   |-- train.csv
|-- my_code
|   |-- foo.py
|   |-- bar
|   |-- ...
|-- office
|   |-- office.data
|-- README
|-- wine
    |-- wine.data

6 directories, 7 files
$ cat Doe-John-HW2/README
To run my code, open HW2.ipynb and ...
```

We expect the report to contain all of the following parts. The Report should contain the following:

- You must include your code as well to get credit. No code submitted means a zero in the assignment.
- If you are using a library from somewhere else, please mention it here as well. We hope you use Piazza for this so more people benefit.
- Again, do not simply use a library functions that already implement boosting or random forests for you, although you may build from an existing decision tree structure implementation.
- You are encouraged to discuss and help others with anything short of giving them your code. There are many references on-line, especially with MNIST. However, you MAY NOT use code from the Internet.

Suggestions

While examining the results from the two different types of random forests, you may observe the version with unlimited depth performing poorly in test error. Try reasoning why and generate some figures to backup your hypothesis. Also, while examining error rates over the number of estimators in a Boosted ensemble, you notice an effect of diminishing returns or a plateauing of performance. Again, try derive hypothesize and show why it would be the case.