CS57300: Homework 4

Due date: Wednesday April 12, midnight (submit via turnin)

Ensembles

In this assignment you will run further experiments with the classification task from HW2/HW3.

Consider the following setup:

- Data: Use the `yelp_data.csv` dataset (which we refer to as D) with 2000 reviews.
- Features: Compute binary word features as before; discard the top 100 words, use **1000** features unless otherwise specified (i.e., 101-1100 most frequent words).
- Class label: Use the same class label *isPositive*.
- Evaluation: Use the same evaluation measure: zero-one loss.
- Use the following proportions of D as training set sizes: $[0.025, 0.05, 0.125, 0.25]$.
- Use *incremental* 10-fold cross validation to compute learning curves with training sets of varying size, but constant test set size.
- Report the average performance (e.g., $L_{0/1}$) over the ten-fold cross validation trials and the *standard error*.

New for this assignment:

- Implement decision trees, bagging, and random forests to compare to the Linear SVM from HW3.
    1. Decision trees: Use gini-gain as a score function, grow full trees using a depth limit of 10 and an example limit of 10 (i.e., stop growing when either the depth of the tree reaches 10 or the number of example in a node is fewer than 10).
    2. Bagging: Learn 50 trees, use sampling with replacement to construct pseudosamples.
    3. Random forest: Learn 50 trees, use sampling with replacement to construct pseudosamples, use $\sqrt{p}$ to downsample the features at each node of the tree (where p is the total number of features).

Programming assignment

You should implement your solution using Python. You can use supporting libraries like numpy, scipy as before, but you may not use scikit-learn. As before, you should submit your typed HW report as a pdf along with your source code file.

Code (10 pts)

Name your file hw4.py. Your python script should take three arguments as input.

1. *trainingDataFilename*: corresponds to a subset of the Yelp data (in the same format as `yelp_data.csv`) that should be used as the *training set* in your algorithm.
2. *testDataFilename*: corresponds to another subset of the Yelp data (again in the same format) that should be used as the *test set* in your algorithm.
3. *modelIdx*: an integer to specify the model to use for classification (DT= 1, BT= 2, RF= 3, where DT refers to decision trees, BT refers to bagging, and RF refers to random forests).

Use the same output format as HW3, appending the model name to the score. E.g.:

```
ZERO-ONE-LOSS-DT 0.3106
ZERO-ONE-LOSS-BT 0.3099
ZERO-ONE-LOSS-RF 0.2947
```

Note that your submitted code should include both the basic code described above to test your models from the command line and the code you use for the analysis below.

Analysis (40 pts)

1. *Assess whether ensembles improve performance.*
   (a) Plot the learning curves for the three models plus SVM (in the same plot), including error bars that indicate ± 1 standard error, from the evaluation based on incremental CV.
   (b) Formulate a hypothesis about the performance difference you observe for one of the ensembles compared to the SVM. Discuss how the observed data support the hypothesis (i.e., are the observed differences significant).
2. *Assess whether the number of features affects performance.*
   Fix the training set size at 500 (0.25%) and vary the number of features: $[200, 500, 1000, 1500]$.
   (a) Plot the learning curves for the three models plus SVM (in the same plot), including error bars that indicate ± 1 standard error, from the evaluation based on incremental CV.
   (b) Formulate a hypothesis about the performance difference you observe for one of the ensembles compared to the SVM. Discuss how the observed data support the hypothesis.
3. *Assess whether the depth of the tree affects performance.*
   Fix the training set size at 500 and vary the depth limit on the decision trees: $[5, 10, 15, 20]$.
   (a) Plot the learning curves for the three tree models (in the same plot), including error bars that indicate ± 1 standard error, from the evaluation based on incremental CV.
   (b) Formulate a hypothesis about the performance difference you observe between two of the models. Discuss how the observed data support the hypothesis.
4. *Assess whether the number of trees affects performance.*
   Fix the training set size at 500 and vary the number of trees in the ensembles: $[10, 25, 50, 100]$.
   (a) Plot the learning curves for the ensemble models (in the same plot), including error bars that indicate ± 1 standard error, from the evaluation based on incremental CV.

(b) Formulate a hypothesis about the performance difference you observe for one of the ensembles compared to the single decision tree. Discuss how the observed data support the hypothesis.

5. *Prove that the expected squared loss for a single example can be decomposed into bias/variance/noise.* Show the decomposition, and identify the bias, variance, and noise terms.

Bonus (15 pts)

Implement boosted decision trees, using the same parameters as for bagging (i.e. same depth limit, same number of trees). Include boosting results in all the experiments above. Formulate at least two hypotheses w.r.t. your boosting results: (1) compare boosting to SVMs, and (2) compare boosting to one of the other ensembles. Discuss how the observed data support the hypothesis.

Submission Instructions:

After logging into data.cs.purdue.edu, please follow these steps to submit your assignment:

1. Make a directory named $'yourName\_yourSurname'$ and copy all of your files there.
2. While in the upper level directory (if the files are in /homes/neville/jennifer_neville, go to /homes/neville), execute the following command:

   turnin -c cs57300 -p HW4 your_folder_name

   (e.g. your prof would use: turnin -c cs57300 -p HW4 jennifer_neville to submit her work) Keep in mind that old submissions are overwritten with new ones whenever you execute this command.

   You can verify the contents of your submission by executing the following command:

   turnin -v -c cs57300 -p HW4

   Do not forget the -v flag here, otherwise your submission will be replaced with an empty one.

Your submission should include the following files:

1. The source code in python, named `hw4.py`.
2. Your evaluation & analysis in named `report.pdf`. Note that your analysis should include learning curve graphs as well as a discussion of results.
3. A README file containing your name, instructions to run your code and anything you would like us to know about your program (like errors, special conditions, etc).