CS534 — Implementation Assignment 3 — Due 11:59PM Nov 28, 2018

General instructions.

1. The following languages are acceptable: Java, C/C++, Python, Matlab

2. You can work in a team of up to 3 people. Each team will only need to submit one copy of the source code and report. You need to explicitly state each member's contribution in percentages (a rough estimate).

3. Your source code and report will be submitted through TEACH

4. You need to submit a readme file that contains the programming language version you use (e.g. python 2.7) and the command to run your code (e.g. python main.py).

5. Please make sure that you can be run code remotely on the server (i.e. babylon01) especially if you develop your code using c/c++ under visual studio.

6. Be sure to answer all the questions in your report. You will be graded based on your code as well as the report. In particular, **the clarity and quality of the report will be worth 10 pts**. So please write your report in clear and concise manner. Clearly label your figures, legends, and tables.

7. In your report, the results should always be accompanied by discussions of the results. Do the results follow your expectation? Any surprises? What kind of explanation can you provide?

Decision Tree Ensemble for Optical Character Recognition (total points: 80 pts + 10 report pts + 10 result pts)

In this assignment we continue working on the Optical Character Recognition task to classify between two numbers 3 and 5. The goal in this assignment is to develop variations of the **decision tree**.

Data. The data for this assignment is generated from the data of implementation assignment 2. We apply the Principal Component Analysis (PCA) to reduce the dimension from 784 to 100. Here is a short description of each train and validation split:

(a) **Train Set (pa3_train_reduced.csv):** Includes 4888 rows (samples). Each sample is in fact a list of 101 values. The first number is the digit's label which is 3 or 5. The other 100 floating values are the the flattened feature values given by the PCA.

(b) **Validation Set (pa3_valid_reduced.csv):** Includes 1629 rows. Each row obeys the same format given for the train set. This set will be used to tune the hyper parameters and select the best model.

Important Guidelines. For all parts of this assignment:

(a) Please assign labels +1 to number 3 and -1 to label 5.

(b) Please do not add bias to the features.

Part 1 (20 pts) : Decision Tree (DT). For this part we are interested in using a decision tree with below configuration:

- The DT uses gini-index to measure the uncertainty. Specifically if we have a node split from list A to two left and right lists AL and AR as depicted in figure 1 then

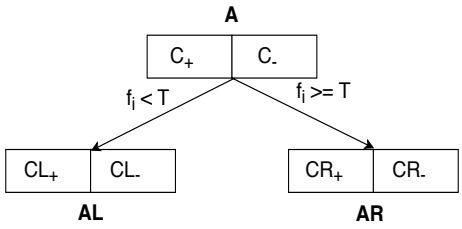

Figure 1: Split list A to two lists AL and AL with feature f_i at threshold T

the benefit of split for feature f_i at threshold T is computed as:

$$B = U(A) - p_l U(AL) - p_r U(AR) \tag{1}$$

Where U is the gini-index function which is computed for a list such as AL as follows:

$$U(AL) = 1 - p_+^2 - p_-^2 = 1 - (\frac{CL_+}{CL_+ + CL_-})^2 - (\frac{CL_-}{CL_+ + CL_-})^2 \tag{2}$$

and p_l and p_r are the probabilities for each split which is given by

$$p_l = \frac{CL_+ + CL_-}{C_+ + C_-} \tag{3}$$

- The feature values are continuous. Therefore you need to follow the descriptions and hints given in the slide to search for the best threshold T for a feature f_i.

Please implement below steps:

(a) Create a decision tree with maximum depth of 20 (root is at depth=0) on the train data. (Note that a normal implementation of the tree should take about 220 seconds on Babylon for this maximum of depth.)

(b) Using the created decision tree, compute and plot the train and validation accuracy versus depth.

(c) Explain the behavior of train/validation performance against the depth. At which depth the train accuracy reaches to 100% accuracy? If your tree could not get to 100% before the depth of 20, keep on extending the tree in depth until it reaches 100% for the train accuracy.

(d) Report the depth that gives the best validation accuracy?

Part 2 (30 pts) : Random Forest (Bagging). In this part we are interested in random forest which is a variation of bagging without some of its limitations. Please implement below steps:

(a) Implement a random forest with below parameters:
n : The number of trees in the forest.
m : The number of features for a tree.
d : Maximum depth of the trees in the forest.

Here is how the forest is created: The random forest is a collection of n trees. All the trees in the forest

has maximum depth of d. Each tree is built on a data set of size 4888 sampled (with replacement) from the train set. In the process of building a tree of the forest, each time we try to find the best feature f_i to split, we need to first sub-sample (without replacement) m number of features from 100 feature set and then pick f_i with highest benefit from m sampled features.

(b) For $d = 9$, $m = 10$ and $n \in [1, 2, 5, 10, 25]$, plot the train and validation accuracy of the forest versus the number of trees in the forest n.

(c) What effect adding more tree into a forest has on the train/validation performance? Why?

(d) Repeat above experiments for $d = 9$ and $m \in [20, 50]$. How greater m changes the train/validation accuracy? Why?

Part 3 (30 pts) : AdaBoost (Boosting). For this part we are interested in applying AdaBoost to create yet another ensemble model with decision tree. Considering the AdaBoost algorithm described in the slide, please do below steps:

(a) Let the weak learner be a decision tree with depth of 9. The decision tree should get a weight parameter D which is a vector of size 4888 (train size). Implement the decision tree with parameter D such that it considers D in its functionality.
(Hint: It changes the gini-index and also the predictions at leaves).

(b) Using the decision tree with parameter D implemented above, develop the AdaBoost algorithm as described in the slide with parameter L.

(c) Report the train and validation accuracy for $L \in [1, 5, 10, 20]$.

(d) Explain the behavior of AdaBoost against the parameter L.

Submission. Your submission should include the following:
1) Your source code with a short instruction on how to run the code in a readme.txt.
2) Your report only in pdf, which begins with a general introduction section, followed by one section for each part of the assignment.
3) Please note that all the files should be in one folder and compressed only by .zip.