CSL603– Machine Learning
Lab 1

Due on 28/8/2018 11.55pm

Instructions: Upload to your moodle account one zip file containing the following. Please do not submit hardcopy of your solutions. In case moodle is not accessible email the zip file to the instructor at ckn@iitrpr.ac.in. Late submission is not allowed without prior approval of the instructor. You are expected to follow the honor code of the course while doing this homework.

1. **You are to work individually for this lab.**
2. This lab may be implemented using Java, C++, C, or Python.
3. A neatly formatted PDF document with your answers for each of the questions in the homework. You can use latex, MS word or any other software to create the PDF.
4. Include a separate folder named as 'code' containing the scripts for the homework along with the necessary data files. Ensure the code is documented properly.
5. Include a README file explaining how to execute the scripts.
6. Name the ZIP file using the following convention rollnumberhwnumber.zip

---

Decisions Trees and Forests

This homework will help you gain a better understanding of decision tree based inductive classifier. You will be implementing the ID3 decision tree algorithm that we discussed in the class. The goal is to predict the sentiment of a movie review. We will be using the Large Movie Review Dataset from Stanford for running our experiments. The core dataset contains 50,000 reviews split into train and test sets. This is a large dataset and therefore to make it easier to handle, we will only work with a random subset of 1000 instances from the dataset. We will use the labeledBow.feat files in the train and test directories of the dataset. Each line in this file is a tokenized bag of words features that were used by the creators of the dataset in their experiments. This file follows an ascii-sparse vector format, where feature indices start from 0. An entry of 0:7 in this file means that the first word in the imdb.vocab file appeared 7 times in the review. Each line in the file refers to a single review. The first entry in the line is the rating of the review on a scale of 1-10. We will also consider positive reviews to have ratings >= 7 and negative reviews to have a rating <= 4. Download the dataset from the following reference [2].

The important step of the ID3 algorithm involves choosing an attribute for creating the decision function at every node in the decision tree. Use information gain as the measure to select the *best* attribute for splitting a node.

1. Create a training set containing a random sample of 1000 observations, and a test set containing the 1000 instances from the train and test directories. Ensure that there are equal number of positive and negative instances in both the train and test sets. We will sample the data points at random to obtain our mini train and test sets only once. This random subset will then be used to run all the experiments. This way, we will be able to perform experiments that can be reproduced with the same sample.
2. Use the training set to learn a decision tree. Discuss the statistics of the learned tree, for example, effect of early stopping on the number of terminal nodes, effect of early stopping on the prediction accuracy on the training dataset and the left out test dataset, attributes that are most frequently used as split functions in the internal nodes of the tree, etc.
3. Let us add some noise to the dataset and observe its effect on the decision tree. Add 0.5, 1, 5 and 10% noise to the dataset. You can add this noise by randomly switching the label of a proportion of data points. Construct the decision tree for each noise setting and quantify the complexity of the learned decision tree using the number of nodes in the tree. Document and discuss your observations about the quality of the model learned under these noise conditions.
4. Produce a pruned tree corresponding to the optimal tree size by computing the prediction accuracy on the test set. Use a post-pruning strategy to prune the tree that has been learned without applying any early stopping criteria. Document the change in the prediction accuracy as a function of pruning (reduction in the number of nodes or rules/clauses obtained from the tree).
5. Learn a decision forest that uses feature bagging. Use majority voting for predicting the label for a test data point. Study the effect of number of trees in the forest on the prediction accuracy of the test data set. Document and discuss your results.

We should be able to execute your code on institute linux machines in this format.
Executable *filename exptno*
Where **Executable** is the name of the executable file, *filename* is the name of the file containing the data and *exptno* is the experiment number (2-5).

Output Format:
The output of your code should closely match the results presented in your report. The results for the different runs must be grouped together. For example, if for experiment 1, you are determining attributes that are most frequently used as split function in the internal nodes of the tree, the output should closely resemble
Number of times an attribute is used as the splitting function
$x_1 \;\; count_1$
$x_2 \;\; count_2$
$\vdots$
$x_D \;\; count_D$

where *Number of times an attribute is used as the splitting function* is the title of the study, x_1 is the attribute and $count_1$ is the number of times it was used as a splitting function.

Another example is for experiment 5 where you would like to compute the prediction accuracy on the training and test set as a function of number of trees in the decision forest. Here the output should closely resemble

Effect of number of trees in the forest on train and test accuracies

$ntrees_1\ tr - acc_1\ ts - acc_1$

$ntrees_2\ tr - acc_2\ ts - acc_2$

⋮

$ntrees_n\ tr - acc_n\ ts - acc_n$

where *Effect of number of trees in the forest on train and test accuracies* is the title of the study $ntrees_1$ is the initial number of trees in the forest, $tr - acc_1$ and $ts - acc_1$ are the training and test accuracies obtained using this learned decision forest model respectively.

An important aspect of machine learning is reproducibility of the results presented in a paper/report. Therefore, we will run your code to see if the results are closely matching with what you have presented in the report. Any deviation beyond a reasonable threshold will be considered as fudging of results and will invite severe penalty.

Reference

[1] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. (2011). Learning Word Vectors for Sentiment Analysis. The 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011).

[2] http://ai.stanford.edu/~amaas/data/sentiment/index.html