DS 222: ML with Large Datasets (2018)

Assignment 1

Weightage: 10%
Due: 11:59 PM (IST) Sep 8, 2018

In this assignment, you will train a Naive Bayes classifier in both local and MapReduce mode.

- Part 1:
  - Local Naive Bayes: In this part of assignment you will implement Naive Bayes in-memory. You can use either Java or Python for your implementation.
- Part 2:
  - In this part of assignment you will re-implement Naive Bayes on Hadoop MapReduce framework.

Data

We are using a dataset extracted from DBpedia. The labels of the article are mapping-based types of the document. There are in total 50 classes in the dataset, and they are from article labels in the dataset. For more information about this dataset, you can refer to http://wiki.dbpedia.org/Downloads2015-04.

There are multiple class labels per document. This means that there is more than one correct answer for each document. The data for this assignment is available at:

- HDFS: /user/ds222/assignment-1/DBPedia.full/
- Turing local: /scratch/ds222-2017/assignment-1/DBPedia.full

The format is one document per line with following columns separated by a tab character :

- List of labels separated by comma
- Document

The first two tokens in the document field are the names of the entity and a type field. You may ignore these. Each data set appears in full in one file, and is split into a train and test set, as indicated by the file suffix.

It might be best to debug your implementation with a small dataset before running it on the full dataset mentioned above. For this reason, a small dataset with same structure as above is available here:

- /user/ds222/assignment-1/DBPedia.verysmall (HDFS)
- /scratch/ds222-2017/assignment-1/DBPedia.verysmall/ (local)

You can refer to the document here to get familiar with the Turing Cluster to be used for the assignment.

Submission

- This is an individual assignment. Each student should submit individual work
- If you have consulted with anyone or sought help from any source, then please acknowledge that clearly in the report.
- Please use Latex to write your report. You may use the format available here
- All submissions will be done through Github. Please submit your code and report to Github. A Google form for submission will be posted here shortly.
- Please state the training, development, and test accuracies of the local as well as MapReduce-based implementation of the NB algorithm. If your model is able to predict any one of the true labels on a given instance, then you may consider it to be a correct prediction.
- Report the number of parameters in the models.
- Please report on the train and test timings (wall clock time) of the two implementations.
- Set the number of reducers in your hadoop run to 1, 2, 5, 8, 10, and record the wall clock time. (The wall clock time of reducers can be inferred from syslogs–use the time between “map 100% reduce 0%” and “map 100% reduce 100%”). Plot a curve, where the horizontal axis is the number of reducers, and the vertical axis is the wall time. Is the wall time linear with respect to the number of reducers? Explain what you observed.
- Also submit the log file of the controller to Github.

Please post on course mailing list (2018DS222@iisc.ac.in) for any issues you face or clarifications you need regarding Assignment 1.