10-605 Assignment 2: Streaming Naive Bayes

Namit Katariya (andrew id: nkatariy)

February 4, 2013

QUESTION 1

- Output of training and testing on the *abstract.tiny* dataset:

```
[pt,tr,hu,es,ru,pl,ca,nl,sl,fr,ga,de,hr,el]  es -1153.0693283794321
[es,ru,ca,fr,de,el]                          hu -608.7886671488046
[pt,es,ru,pl,fr,de]                          pl -423.69724861996394
[pt,es,ru,fr,de]                             hr -1284.964848739909
[es,pl,fr,de]                                sl -1196.4653958490803
[pl,ca,de]                                   hr -166.0102146311306
[pt,de]                                      sl -526.8324405049511
[pt,es,pl,fr,de]                             nl -694.8456461467524
[es,ru,de]                                   el -412.7442462411836
[fr,de]                                      hr -761.3470264877526
[nl]                                         hu -320.78872549292925
[fr,ga]                                      pl -171.56168490813403
[hu]                                         el -412.22099809741906
[fr]                                         sl -942.251046966511
[fr,de]                                      hr -167.2141874354565
[pt,es,pl,ca,nl,fr]                          nl -56.24669699411067
[el]                                         nl -61.48047583952113
[de]                                         hu -728.5117534282172
[fr]                                         nl -36.19910766605365
[pt,fr]                                      fr -102.17297364515815
[pl]                                         pl -140.47162703358435
[ru,pl,de]                                   hu -150.4372995050212
[fr]                                         pt -41.56044405900344
[pt,fr]                                      pt -132.2834942841487
[sl,fr,de]                                   hu -311.76167589728857
[nl]                                         ca -246.0775024056901
[ru,pl]                                      el -634.5806969564671
[pt,tr,es,ru,pl,ca,nl,fr,de,el]              pt -170.81695352609407
[pt,es,ru,nl,fr,de]                          pt -156.26694453558
```

Dataset	Percent Correct
abstract.tiny	27.59
abstract.small	64.69
abstract.full	71.50
links.tiny	25.0
links.small	36.08
links.full	71.50

QUESTION 2.A

In terms of accuracy, I got around the same numbers for both the datasets. However, links took a lot more time than abstract. Another thing was that I couldn't run streaming NB on the links dataset without breaking it up into multiple files whereas abstract could be run as is. The tokens in links were much longer (long set of underscore-separated words) so the neededWords table will be larger and the look-up to build the counts table might be longer. This might explain the longer execution time.

QUESTION 2.B.I

One could build a classifier for each level in the tree (hierarchy) but allow a lower level classifier to predict positive for a node n only if the classifier for the level above also predicts positive for parent(n) or in other words, we use the child classifier only if the parent classifier predicted it as negative. For training on lower levels, one could use only instances in the training set that belong to the corresponding parent.

QUESTION 2.B.II

I think the vocabSize that we use for smoothing a particular set of classes should be the size of vocabulary calculated only from the documents belonging to their parent class node. These probabilities will be relatively higher than if we had used the total vocabulary size and so we might get better results.

BONUS QUESTION

Timing Comparison

Assignment 1	8.002 sec
Assignment 2	80.565 sec

Commands used : I created the following .sh files and measured time using *time bash a1.sh* and *time bash a2.sh*. User+System time was measured and divided by 10 to get the average execution time.

```
Assignment 1: a1.sh
------------------------
for i in 1 2 3 4 5 6 7 8 9 10
do
    cat ../asgn1_data/RCV1.small_train.txt | java NBTrain > /dev/null
done

Assignment 2: a2.sh
--------------------
for i in 1 2 3 4 5 6 7 8 9 10
do
    cat ../asgn1_data/RCV1.small_train.txt | java -Xmx128m NBLargeVocabTrain |
                          sort -k1,1 -t ';' -T . | java -Xmx128m Aggregator > /dev/null
done
```