BigML Assignment 2: Small Memory Footprint Streaming Naive Bayes

Runyun Zhang
runyunz@andrew.cmu.edu

1. **Compare and discuss the performance for the full links vs full abstract data.**
   The difference of links data and abstract data is that links data has a far smaller vocabulary size, but every token/term is far longer than a single word in abstract data.
   In this way, although links data output less records in total, it is more likely for links data to use up the memory of the count hash table, which requiring more times to flush hash table and output temporary counts. From my running record on small dataset, processing links dataset took **twice** as long time as abstract dataset during each steps: training, sorting, merging, and testing.

```
Runyuns-MacBook-Pro:Hw2 runyunz$ cat abstract.small.train | time java -Xmx128m NBTrain | time
sort -k1,1 | time java -Xmx128m MergeCounts | time java -Xmx128m NBTest abstract.small.test >
nbtest.res
      18.83 real         4.08 user         1.22 sys
      30.46 real        21.49 user         1.47 sys
      30.49 real         8.08 user         1.32 sys
      32.51 real         8.01 user         2.84 sys
Runyuns-MacBook-Pro:Hw2 runyunz$ cat links.small.train | time java -Xmx128m NBTrain | time
sort -k1,1 | time java -Xmx128m MergeCounts | time java -Xmx128m NBTest links.small.test >
nbtest.res
      47.22 real         8.23 user         2.37 sys
      69.81 real        56.16 user         2.96 sys
        69.83 real        16.69 user         2.85 sys
        73.56 real        16.20 user         5.99 sys
```

   One way to improve this case for link dataset might be instead of use link directly, we can create a mapping and use index of each link as the training feature, which would reach a better compression.

2. **Using a local copy of the RCV1.small train.txt file from Assignment 1, compare the performance of creating your Naive Bayes feature dictionary for last assignment and this assignment. Time all parts of the dictionary creation (including, for example, sorting and combining counts for your Assignment 2 solution). Average over 10 calls. Please include in your write up the commands you used to do this comparison.**

Without considering the process of loading dictionary in NBTest process, for Hw1, the dictionary is created just during NBTrain process.

The test script is:

```
for i in {1..10}
do
 cat RCV1.small_train.txt | time java –Xmx128m NBTrain > res
done
```

The result is:

```
10.93user 2.37system 0:08.53elapsed 155%CPU (0avgtext+0avgdata 456304maxresident)k
0inputs+6840outputs (0major+11761minor)pagefaults 0swaps
```

Without considering the process of loading dictionary in NBTest process, for Hw2, the dictionary is created during NBTrain, Sort, and MergeCount process. Depending on the setting of hash table size in NBTrain, the results may vary.

The test script is:

```
for i in {1..10}
do
     cat RCV1.small_train.txt | time java –Xmx128m NBTrain | time sort –k1,1 | time java –Xmx128m MergeCounts > merge.res
done
```

Here is a result of setting Hash Table size to 40,000:

NBTrain:

```
0inputs+63440outputs (0major+773minor)pagefaults 0swaps
37.54user 24.94system 1:11.79elapsed 87%CPU (0avgtext+0avgdata 314288maxresident)k
```

Sort:

```
0inputs+63608outputs (0major+12928minor)pagefaults 0swaps
17.68user 2.53system 0:17.16elapsed 117%CPU (0avgtext+0avgdata 357696maxresident)k
```

MergeCount:

```
0inputs+64outputs (0major+15603minor)pagefaults 0swaps
17.35user 1.51system 1:02.95elapsed 29%CPU (0avgtext+0avgdata 204480maxresident)k
```

3. **How can we get the most informative features of a specific class given the trained naive Bayes model? In other words, if we are interested in figuring out the most predictive features for a class of wikipedia articles (e.g. German), what can we do? This is an open question. (5 points)**

For a specific class, find those words with large counts could help, i.e. Count (W=w, Y=y). However, since common words are likely to spread over different class, it is worth considering to adopt a similar strategy as TFIDF, that is, to compute the term frequency of a specific class versus the term frequency of all the classes.

4. **Answer the questions in the collaboration policy on page 1.**

Did you receive any help whatsoever from anyone in solving this assignment? No.

Did you give any help whatsoever to anyone in solving this assignment? No.