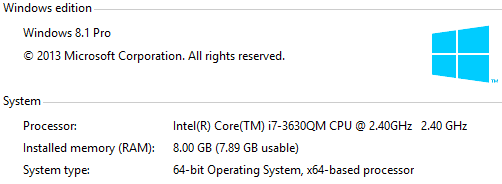
MMD Assignment 1- Sanath Bhat

Note: The regular algorithm had to be modified slightly to be able to prevent out of memory issues. All words with count=1 after the merge operation were thrown away, since the hashmap was getting so big with all the words in the vocabulary that it wouldn’t fit entirely in the limited amount of memory.

1. Testing results:

System specs:



Some Results:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Data Set** | | **Buffer Size in NBTrain** | **Classification Accuracy** | **Single-Run Runtimes (seconds)** | |
| **Training** | **Testing** | **Training** | **Classification** |
| Very Small | Very Small | 0 | 75% | 0.636 | 0.13 |
| Very Small | Very Small | 10K | 75% | 0.418 | 0.128 |
| Small | Small | 0 | 92.7% | 45.487 | 1.824 |
| Small | Small | 10K | 92.7% | 18.65 | 1.624 |
| Full | Full | 0 | 94.88 | 5681.11 | 12775.48\* |
| Full | Full | 100K | 94.88% | 881.547 | 9364.227\* |
| Very Small | Small | 0 | 71.53% | 0.678 | 1.616 |
| Small | Full | 0 | 92.84% | 32.505 | 45.181 |

\* Classification time can be reduced drastically by writing results to a file instead of writing to console output. These numbers could also be affected by other processes running on that computer as the CPU was at 80-100% utilization constantly.

1. Buffered Message Sizes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Training Set** | **Total Buffered Messages’ Size(MB)** | | | |
| **10** | **100** | **1000** | **10000** |
| Very Small | 0.405 | 0.31 | 0.215 | 0.143 |
| Small | 42.817 | 33.18 | 23.58 | 15.38 |
| Full | 4251.75 | 3257.43 | 2238.8 | 1506.05 |

1. & 4.

Need for smoothing:

When calculating Pr(y’,x1,x2, …,xn) term, if a word xk was never found under a particular class label during training, then C(X=xk ^ Y=y’) = 0 which causes Pr(y’,x1,x2, …,xn) to become zero implying a zero probability of a document to have a label y’ just because of a word which was never found under that label when training. Thus we need to smoothing to prevent such a undesirable effect which would maintain Pr(y’,x1,x2, …,xn)>0 under any circumstances.

Effect of smoothing on RCV1\_small dataset: (Buffer size = 1000)

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Without smoothing** | **With Laplacian smoothing** | **With Dirichlet smoothing(For Q4)** |
| **Accuracy** | 2.35% | 92.7% | 89.98% |
| **Time (secs)\*** | 1.243 | 1.2 | 1.216 |

\*Time taken for classification only

5. To predict all the labels of a document after calculating the log probabilities for each label in *dom(labels)* we simply do an additional step. We compute the average of the log probabilities of each class and take that as the threshold. Now, if the log probability of a label is greater than the threshold, then the document has that label. This way a document can have at least one label (When one of the labels is much greater than the others causing the threshold to be pulled towards itself) and at most all the labels in *dom(labels)* (In case of equal log probabilities for all labels).

Summarized is the above:

∀, if log(Pr(y,x1,x2,…xn) > Threshold, then y is a label for d