DS222 - Assignment 1

Nashez Zubair [*1]

1. Number of Parameters

Number of Parameters in the models are $vocabularySize*$ 50 since we are using a bernoulli model. The observed number of unique words for our models are = 483243 and the number of classes are given to be 50. Thus total number of parameters observed are 2416215.

2. Implementations

2.1. Preprocessing

The words were split on the spaces and any punctuation or special characters were removed. For example, the word *murphy's* in the text would become *murphys*. The labels were kept as it is.

2.2. Local Mode

The local mode java naive bayes program has been implemented and runs as expected with accuracies given in Table 1 and runtimes given in 2.

2.3. MapReduce Mode

The mapreduce implementation of the code has been implemented upto the training phase and runs correctly. I have not been able to implement the testing phase code and thus I do not have the accuracies. But I do have the training runtimes of these jobs for all number of reducers used shown in Table 2.

3. Observations

Table 1. Accuracy Observations

	Training	Development	Testing
Local	73.26	54.99	59.61
MapReduce			

The wall time as observed in Figure 1 is not linear with the number of reducers. We observe that it decreases first, reaching a minimum and then starts increasing. We then present a more detailed graph in Figure 2 where we observe the different components of the reducer time such as the shuffle time, merge time and the reduce time (time taken by just the actual reduce operation). We observe that the reducer time is still decreasing but the shuffle time has increased which has caused the wall clock time interval between map completion and reduce completion to increase. The decrease first is explained by the fact that we are giving it more number of machines and hence can be explained by data parallelism. But as number of reducers grow, the shuffling overheads outweigh the benefits of data parallelism, in other words it becomes costlier to distribute data rather than just computing it. Thus the time increases.

Table 2. Runtime Observations(s)

	Training	train.txt	devel.txt	test.txt
Local	18	535	103	42
MapReduce1	228			
MapReduce2	208			
MapReduce5	193			
MapReduce8	189			
MapReduce10	213			

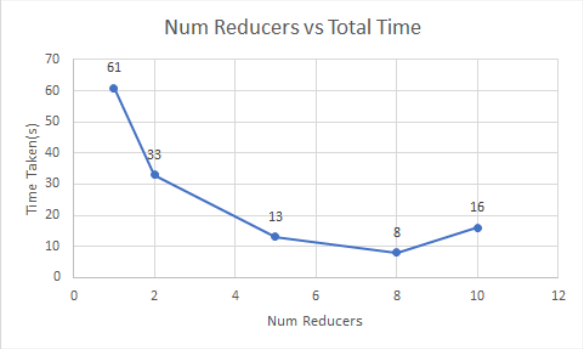

Figure 1. Effect of increasing number of Reducers on Runtime

[*]Equal contribution [1]M.Tech, CDS. Correspondence to: <nashezzubair@iisc.ac.in>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017.

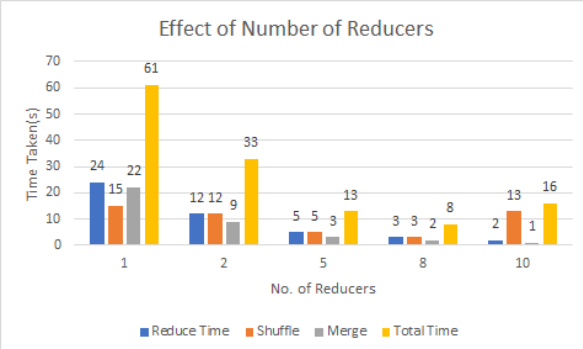

Figure 2. Detailed Effect of increasing number of Reducers: Different Times taken