Assignment 4

Versions of frameworks

Scala: 2.11.0

Spark: 2.3.1

Running of the program

The jar file produces a output json file (in the folder where the command to run the code is executed) using the Clustering algorithms – KMeans, Bisecting KMeans based on the input parameters passed as arguments to the file. In order to run the file the following commands should be used:

spark-submit –class Task1 Utkarsh_Gera_Clustering.jar <input file path> <feature> <N> <Iteration>

spark-submit –class Task2 Utkarsh_Gera_Clustering.jar <input file path> <algorithm> <N> <Iteration>

As mentioned in the problem description the output file contains the clusters details with the algorithm type used to generate them, WSSE and the cluster object containing the cluster details like id, top 10 words etc. Ex:

```
{
     "algorithm":"K-Means",
     "WSSE":0.52,
     "clusters":[ {
          "id":1,
          "size":50,
          "error":0.25,
          "terms": ["restaurant","food","good"…]
     },
….

}
```

Since I used the third party library json4s it dumps the json into file in single line keeping the json structure intact. (Indents don't matter and not GMM implementation refer post 101 by Ruobo Jiang)

The Task1 implements the KMeans and takes around 15 mins for both Word count as well as the TFIDF as a feature. It also predicts like the mllib WSSE, size of a cluster, error within a cluster and top 10 words of a cluster. It converts the reviews into TF or TFIDF vectors depending upon the feature mentioned and then calls the self-implemented version of KMeans that uses the capability of the vector data structure to calculate the mean of RDD of vectors and the square distance between 2 vectors.

Task2 utilizes the same structure as the Task1 except for the self-written code for KMeans is replaced by the code provided by mllib library for the KMeans or Bisecting Kmeans depending upon the arguments provided.

If the program finds any issue in the arguments or you don't provide these 4 arguments/provide more than 4 arguments the program will exit with a proper message like:

Need at four arguments - Input file, feature, number of clusters and max number of iterations

or

Need at four arguments - Input file, algorithm, number of clusters and max number of iterations

Explanation of the Implementation

KMeans or Bisecting KMeans are the clustering algorithms that classify points to cluster based on the Euclidian distance. Self-implementation of KMeans does the same job as the KMeans algorithm provided by mllib. It has a predict function that takes in the list of clustroids and the Vector to categorize to the cluster and returns the cluster id depending upon its index number in the array of clustroids.

In the mllib KMeans or Bisecting KMeans we get a model which can be reused to classify the vectors again to their respective cluster, so that we can calculate the error within cluster, size and the top 10 words. In case of the self-implementation of the KMeans, I return the final clusters which along with predict method can help me to classify the review vectors to the clusters to get the same thing.

Task2 generates output file by the name - "Utkarsh_Gera_KMeans_" + dataset type + "_" + algorithm + "_" + numClusters + "_" + numIterations + "_.json". Since it runs on 2 data sets: large and small, and also for 2 algorithms KMeans and Bisecting KMeans, we have a total of 4 files for output. Task 2 has the numFeatures set to 2^80 for the HashingTF() to achieve a lower error.

Task1 generates output file by the name – "Utkarsh_Gera_KMeans_small_" + feature + "_" + numClusters + "_" + numIterations + ".json". So there are a total of 2 files for Task1, 1 for each feature – word count and TFIDF.

Bonus for Problem 3

Please refer Utkarsh_Gera_Visualization.pdf