Big Data Assignment-1

Hruday Vishal Kanna Anand

1006874517

2. Explain advantages and disadvantages of using K-Means Clustering with MapReduce.

A.

Advantages-

- Map reduce allows parallel computation of a large dataset there by reducing the compute time.
- Map reduce also enables many commodity computers to work together to perform K-means clustering more efficiently
- Hadoop map reduce also has fault tolerance by creating replicate files.

Disadvantages-

- Map reduce doesn't efficiently handle the iterative component of K-means well, as after every iteration the output file must be written to local storage then read again for the next iteration. This causes the whole process to slow down.
- It is also harder to code as not many external libraries can be used with Hadoop map reduce and only works efficiently with java.

3. Can we reduce the number of distance comparison by applying the Canopy Selection? Which distance metric should we use for the canopy clustering and why?

A.

Yes, we can reduce the total number of complex distance comparisons (expensive distance measure) as we only need to compute the complex distance comparison with the data contained within a canopy and not the entire dataset. This process will be continued for each canopy. This significantly reduces the number of distance comparisons as we don't need to iterate over the whole dataset.

The distance metric used for canopy clustering is inverted index (cheap distance measure). Here we choose 2 distance limits d1 and d2 where d1>d2. A data point is chosen as the center of the canopy and all data points within d1 distance are within the canopy and the data points within d2 distance are not used in further iterations to find other canopies. This distance calculation is much simpler than traditional distance metrics used in clustering algorithms like K-means, Expectation-Maximization or Greedy Agglomerative Clustering hence increasing the computation speed.

4. Is it possible to apply Canopy Selection on MapReduce? If yes, then explain in words, how would you implement it?

A.

Yes, it is possible to implement canopy selection using map reduce. In the map segment we would read the data points from the file and choose a data point as the center of a canopy and create key value pairs for the data points within the distance d1 with the key as the canopy identifier and value being the data point. We can further create another identifier in the value to specify (true/ false) if the point is within d2 distance from the center. All the unclassified points would have a key of zero or so to specify they are yet to be classified.

These key value pairs are then passed on to the reducer which then groups all the values (data points) based on their key and writes this file on to the hdfs.

This file is then read by the mapper while unpacking all the datapoints whether they are already classified with a canopy or not. Again, choosing the center with a data point that doesn't have the second value identifier set to true and repeating the above logic.

This will be repeated till all the points have been assigned a canopy and their second value identifier is set to true.

5. Is it possible to combine the Canopy Selection with K-Means on MapReduce? If yes, then explain in words, how would you do that?

A.

Yes, it is possible. We will have to run 2 classes of the map reduce pair- 2 mapper classes and 2 reducers. One pair will be executing the canopy clustering logic that we have specified above, and the other pair will be running the K-means logic. The output of the first map reduce pair (Canopy clustering) will be used by the second K-means map reduce pair to simulate a 2-stage processing. This restricts the k means algorithm to be performed on individual canopy clusters and not the whole dataset, as the data inputted to the k- means map reduce will already be partitioned into clusters. In the second stage, each canopy cluster chooses 'k' centers and groups the data within a cluster to the closest center- map logic. In the k means reducer- new centers are calculated by averaging all the points with the same center. This iteration is continued until a certain stopping condition (max iteration or no significant change in the centers). This k means algorithm is performed on all the canopy clusters simultaneously during the second map reduce.