K-Means Clustering Algorithm Implementation Using MapReduce

by Vishal Doshi

The MapReduce application is packed in 659391383_ASSIGN2.zip. Extract and import in Eclipse to view the source code.

Class and files:

KMeansFileGenerator.java: used to generate data points at random and random centroids are chosen from data generated. User define datapoint generated with 4 datapoint chosen to be centroids at random.

KMeansHadoop: Driver class for the MapReduce Application

KMeansMapper: Mapper maps datapoints to closest cluster

KMeansReducer: Reducer recalculate the centroids and writes them

KMeansPartition: Partitioner assigns reducer according to the cluster id.

Instructions to run:

Step 1: Copy KMeansFileGenerator.java to desktop. Compile and run.

- **javac KMeansFileGenerator.java**
- **java KMeansFileGenerator.**

Step 2: Generate datapoints.txt and centroid_1.txt and move it on Desktop

Step 3: Create ‘kmeansInput’ and ‘centroid’ directory in HDFS

- **./bin/hadoop fs -mkdir kmeansInput**
- **./bin/hadoop fs -mkdir centroid**

Step 4: Move the files from to Desktop to HDFS

- **./bin/hadoop fs -put ~/Desktop/datapoints.txt kmeansInput**
- **./bin/hadoop fs -put ~/Desktop/centroid_1.txt centroid**

Step 5: Export the jar file to desktop and Run .jar file

- **./bin/hadoop jar ~/Desktop/kmeanshadoop.jar KMeansHadoop kmeansInput kmeansOutput/out centroid**

[3 parameters - 1: input directory, 2: output directory, 3: centroid file location]