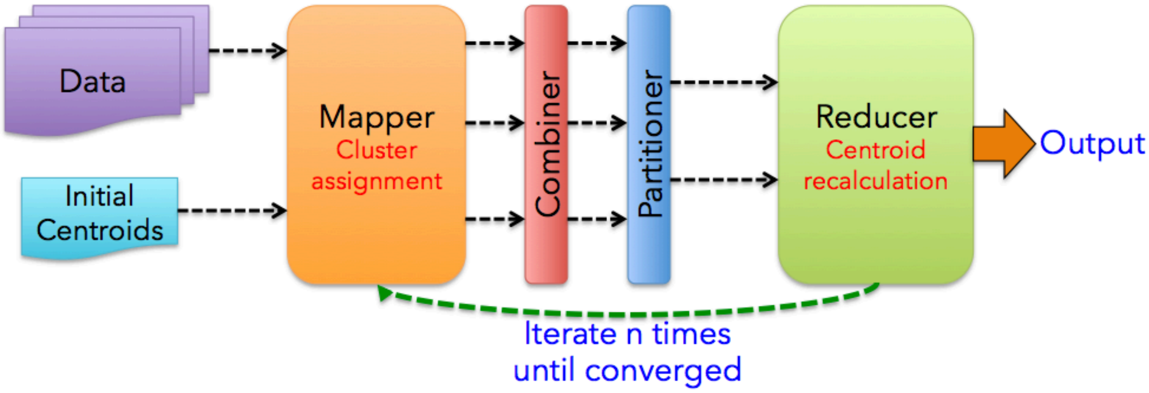
K-Means Clustering Algorithm Implementation Using MapReduce

Instructor: Kunpeng Zhang (kzhang6@uic.edu)
TA: Minghong Xu (mxu29@uic.edu)

The Deadline: 6:00PM, Mar. 2, 2015 (Monday)

The K-Means algorithm is to cluster data points into different partitions based on some distance measures. In this assignment, you need to implement K-Means algorithm using MapReduce to make it scalable. The inputs of the algorithm are the data file and the centroid file. The data file contains *m* data points. Each data point is represented as an *n*-dimensional numeric vector. The centroid file has K data points that are randomly chosen from the data file. The distance measure used here is Euclidian distance. The algorithm flow is described in the following figure.



Data never changes in the MapReduce process

Requirements:

- You need to create an input file containing at least 1000 lines (at least 1000 data points). Each line starts with d_i and a tab followed by 10 numeric values, separated by commas.

  d_i $X_{i1},X_{i2},X_{i3},X_{i4},X_{i5},X_{i6},X_{i7},X_{i8},X_{i9},X_{i10}$

- Your program should be able to handle the input in this way: the key is the d_i and the value is the 10 numeric values

- K here must be in the range of [3, 8]

- You need to implement a customized Partitioner class. The partition function can be implemented by your own design, but returns at least 3 partitions.

- You need to create a customized Combiner class to aggregate your intermediate results from Mapper.

- The stopping criterion is the convergence of the centroids, meaning that the change of centroids between current iteration and last iteration is less than a very small value (e.g., 0.01). DO NOT use the number of iterations as the stopping criterion.

- You need to save all intermediate centroids generated from each iteration in your code. The file name should contain iteration number. For example, the initial centroid file name may be centroid_0.txt. After the first iteration, you will generate centroid _1.txt, after the second iteration, you will generate centroid _2.txt, … (1, 2, … would represent the iteration number).

- **Your codes must be readable and clean.**

- When you submit your codes through blackboard, you need to put all source codes (.java files, NOT jar files), the input file, and some other optional files (e.g., a readme file) into one folder and name that folder as <YOUR UID>_ASSIGN2. Assignments not following this rule will not be graded. In addition, no resubmission after TA grades it. Late submission rule: 10% deduction for one day late. Late submission over a week is NOT acceptable.

DO NOT copy any codes from others. Otherwise, both will be penalized.