Dataset:

|  |  |
| --- | --- |
| 1 | 1 |
| 1.5 | 2 |
| 3 | 4 |
| 5 | 7 |
| 3.5 | 5 |
| 4.5 | 5 |
| 3.5 | 4.5 |

Centroid:

|  |  |
| --- | --- |
| 1 | 1 |
| 5 | 7 |

Iteration 1:

mapper ->

key->1 Val->1,1

key->1 Val->1.5,2

key->1 Val->3,4

key->2 Val->5,7

key->2 Val->3.5,5

key->2 Val->4.5,5

key->2 Val->3.5,4.5

reducer ->

centroid1 = 1.375,1.75

centroid2 = 3.3,4.3

New Centroids saved in the centroids file

Iteration 2:

mapper->

key->1 Val->1,1

key->1 Val->1.5,2

key->2 Val->3,4

key->2 Val->5,7

key->2 Val->3.5,5

key->2 Val->4.5,5

key->2 Val->3.5,4.5

reducer->

centroid 1= 0.8333333333333334,1.0

centroid 2 = 3.25,4.25

New centroids saved in the centroids file in hdfs

Iteration 3:

key->1 Val->1,1

key->1 Val->1.5,2

key->2 Val->3,4

key->2 Val->5,7

key->2 Val->3.5,5

key->2 Val->4.5,5

key->2 Val->3.5,4.5

reducer->

centroid 1= 0.8333333333333334,1.0

centroid 2 = 3.25,4.25

New centroids saved in the centroids file in hdfs (convergence is happening)

Iteration 4:

key->1 Val->1,1

key->1 Val->1.5,2

key->2 Val->3,4

key->2 Val->5,7

key->2 Val->3.5,5

key->2 Val->4.5,5

key->2 Val->3.5,4.5

reducer->

centroid 1= 0.8333333333333334,1.0

centroid 2 = 3.25,4.25

New centroids saved in the centroids file in hdfs (surely convergence is happening)

End of the iteration

final classification in “out” has :

Clusters Instances

1 1,1;1.5,2;

2 3,4;5,7;3.5,5;4.5,5;3.5,4.5;

And the final centroids in the centroid file has:

centroid 1= 0.8333333333333334,1.0

centroid 2 = 3.25,4.25

Pseudo code:

mapper{

save the centroids to a list from centroids file;

for each instance:

find the key as the cluster which is nearest to this instance;

map(key, instance)

}

reducer{

for each instance of a key:

calculate the mean of instances of each cluster;

assign these mean values as the new centroids;

update these centroids in the centroids file

reduce(mean, instance)

}

driverClass{

repeat the job for N times // you decide the N value

// for a random centroid initialization N should be set large enough

}