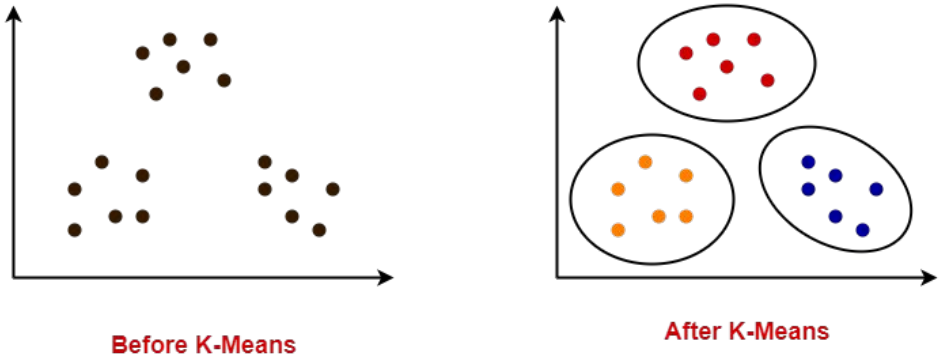
PRACTICAL 7
MapReduce implementation of K-means Clustering algorithm

K-Means Clustering



- K-Means clustering is an unsupervised iterative clustering technique.
- It partitions the given data set into k predefined distinct clusters.
- A cluster is defined as a collection of data points exhibiting certain similarities.

It partitions the data set such that,

- Each data point belongs to a cluster with the nearest mean.
- Data points belonging to one cluster have high degree of similarity.
- Data points belonging to different clusters have high degree of dissimilarity.

Algorithm

K-Means Clustering Algorithm involves the following steps,

Step-01

- Choose the number of clusters K.

Step-02

- Randomly select any K data points as cluster centres.
- Select cluster centres in such a way that they are as farther as possible from each other.

Step-03

- Calculate the distance between each data point and each cluster centre.
- The distance may be calculated either by using given distance function or by using Euclidean distance formula.

Algorithm

Step-04

- Assign each data point to some cluster.
- A data point is assigned to that cluster whose centre is nearest to that data point.

Step-05

- Re-compute the centre of newly formed clusters.
- The centre of a cluster is computed by taking mean of all the data points contained in that cluster.

Step-06

- Keep repeating the procedure from Step-03 to Step-05 until any of the following stopping criteria is met
    - Centre of newly formed clusters do not change
    - Data points remain present in the same cluster
    - Maximum number of iterations are reached

Illustration

Documents (Data Points)	W1 (x-axis)	W2 (y-axis)
D1	2	0
D2	1	3
D3	3	5
D4	2	2
D5	4	6

Illustration

1. calculate the distance between the initial centroid points D2 and D4 with other data points.

Documents (Data Points)	Distance between D2 and other data points	Distance between D4 and other data points
D1	3.17	2.0
D3	2.83	3.17
D5	4.25	4.48

2. group the data points which are closer to centroids.

   Cluster 1: (D1, D4) Cluster 2: (D2, D3, D5)

Illustration

3. calculate the mean values of the clusters created and the new centroid values will these mean values and centroid is moved along the graph.

Clusters	Mean value of data points along x -axis	Distance between D4 and other data points
D1, D4	2.0	1.0
D2, D3, D5	2.67	4.67

New centroid - cluster 1 {D1, D4} is (2.0, 1.0) and
cluster 2 {D2, D3, D5} is (2.67, 4.67)

Illustration

- Iteration II

Documents (Data Points)	Distance between centroid of cluster 1 and data points	Distance between centroid of cluster 2 and data points
D1	1.0	4.72
D2	2.24	2.37
D3	4.13	0.47
D4	1	2.76
D5	5.39	1.89

Clusters	Mean value of data points along x -axis	Distance between D4 and other data points
D1, D2, D4	1.67	1.67
D3, D5	3.5	5.5

cluster 1 (D1, D2, D4) - (1.67, 1.67) and cluster 2 (D3, D5) - (3.5, 5.5)

MapReduce implementation

Classify: Assign observations to closest cluster center

$$z_i \leftarrow \arg\min_j ||\mu_j - \mathbf{x}_i||_2^2$$

set of cluster centers

a data point

map($[\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_k]$, $\mathbf{x}_i$)

$$z_i \leftarrow \arg\min_j ||\mu_j - \mathbf{x}_i||_2^2$$

emit(z_i, $\mathbf{x}_i$)

cluster label

datapoint

$z_i = 2$ (assigned to cluster 2)

datapoint x_i

e.g. emit (2, [17, 0, 1, 7, 0, 0, 5])

MapReduce implementation

Recenter: Revise cluster centers as mean of assigned observations

$$\mu_j = \frac{1}{n_j} \sum_{i:z_i=k} \mathbf{x}_i$$

cluster label (key)

datapoints assigned to cluster j. (have key j)

```
reduce(j, x_in_clusterj : [x1, x3,..., ])
    sum = 0 ← total mass in cluster
    count = 0 ← total # of obs. in cluster
    for x in x_in_clusterj
        sum += x
        count += 1
    emit(j, sum/count)
```

cluster label

total mass / total # obs

MapReduce implementation

Summary of parallel k-means using MapReduce

Map: classification step;
data parallel over data points

Reduce: recompute means;
data parallel over centers