Clustering Big Data

- **Overview**
  - **Clustering** divides a dataset into groups of data items having the same similarity.
  - It requires a **similarity measure.**
    - Distances are normally used to measure the similarity or dissimilarity between two data objects.
    - Example: Euclidean distance:

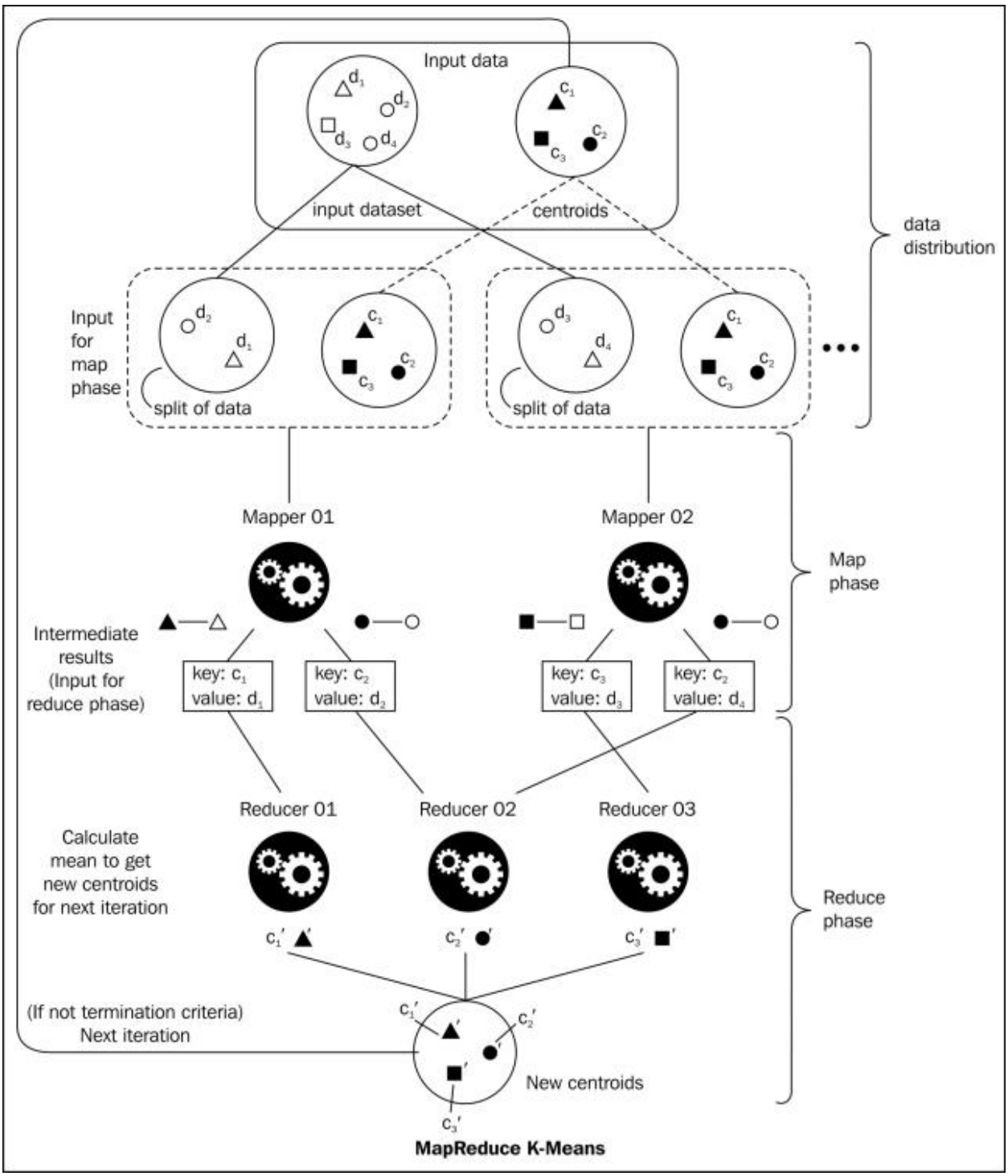
$$distance(i,j) = \sqrt{(\left|x_{i1} - x_{j1}\right|^2 + \left|x_{i2} - x_{j2}\right|^2 + \cdots + \left|x_n - x_{jn}\right|^2)}$$

    - Clustering results are crucially dependent on the measure of similarity (or distance) between "points" to be clustered
  - **A cluster** is a group of data items **similar** (or related) to one another within the same group **dissimilar** (or unrelated) to the data items in other groups.
  - Clustering is an unsupervised learning method:
    - Data items in a dataset are not labelled – do not require predefined classes.
  - Clustering analysis:
    - Given a set of data points, each described by a set of attributes, find clusters such that:
      - Inter-cluster similarity is maximized
      - Intra-cluster similarity is minimized

- **K-Means Clustering**
  - Basic Algorithm:
    - Partition {x1,…,xn} into K clusters-K is predefined
    - Initialization
      - Specify the initial cluster centers (centroids)
        - Iteration until no change
          - Classify:
            - For each object xi
              - Calculate the distances between xi and the K centroids
              - (Re)assign xi to the cluster whose centroid is the closest to xi
          - Re-center: Update the cluster centroids based on the current assignment

  - How to MapReduce K-means? [Apache Mahout Essentials, By Jayani Withanawasam}

- Iteratively, run the MapReduce phase to implement K-Means until the termination criteria is reached - convergence or the number of iterations is reached.
- The Hadoop job splits the dataset into chucks, which are processed by map tasks in a parallel.
- If the following example:
  - The dataset is split across nodes in the cluster by assigning:
    - d1 and d2 to node 1 and
    - d3 and d4 to node 2
  - The map function:
    - Each map function has its data shuck and the list of initial centroids c1, c2, and c3.
    - Compute the distance from each data point in the shuck to all the initial centroids, and the data point is assigned to the closest centroid.
    - The map function outputs **(key = centroid id, value = data point)** pair to the reduce phase.
  - The Reduce function:
    - The data points that belong to a particular centroid are processed in a single node in the reduce phase.
    - Re-center: **new centroid** points are computed using the average of the coordinates of all data points in that cluster.
    - The new centroids are then fed back to the next iteration.


Input data
input dataset
centroids
data distribution
Input for map phase
split of data
split of data
Mapper 01
Mapper 02
Map phase
Intermediate results (Input for reduce phase)
key: c1 value: d1
key: c2 value: d2
key: c3 value: d3
key: c2 value: d4
Reducer 01
Reducer 02
Reducer 03
Calculate mean to get new centroids for next iteration
Reduce phase
(If not termination criteria) Next iteration
New centroids
MapReduce K-Means