Parallelization Example

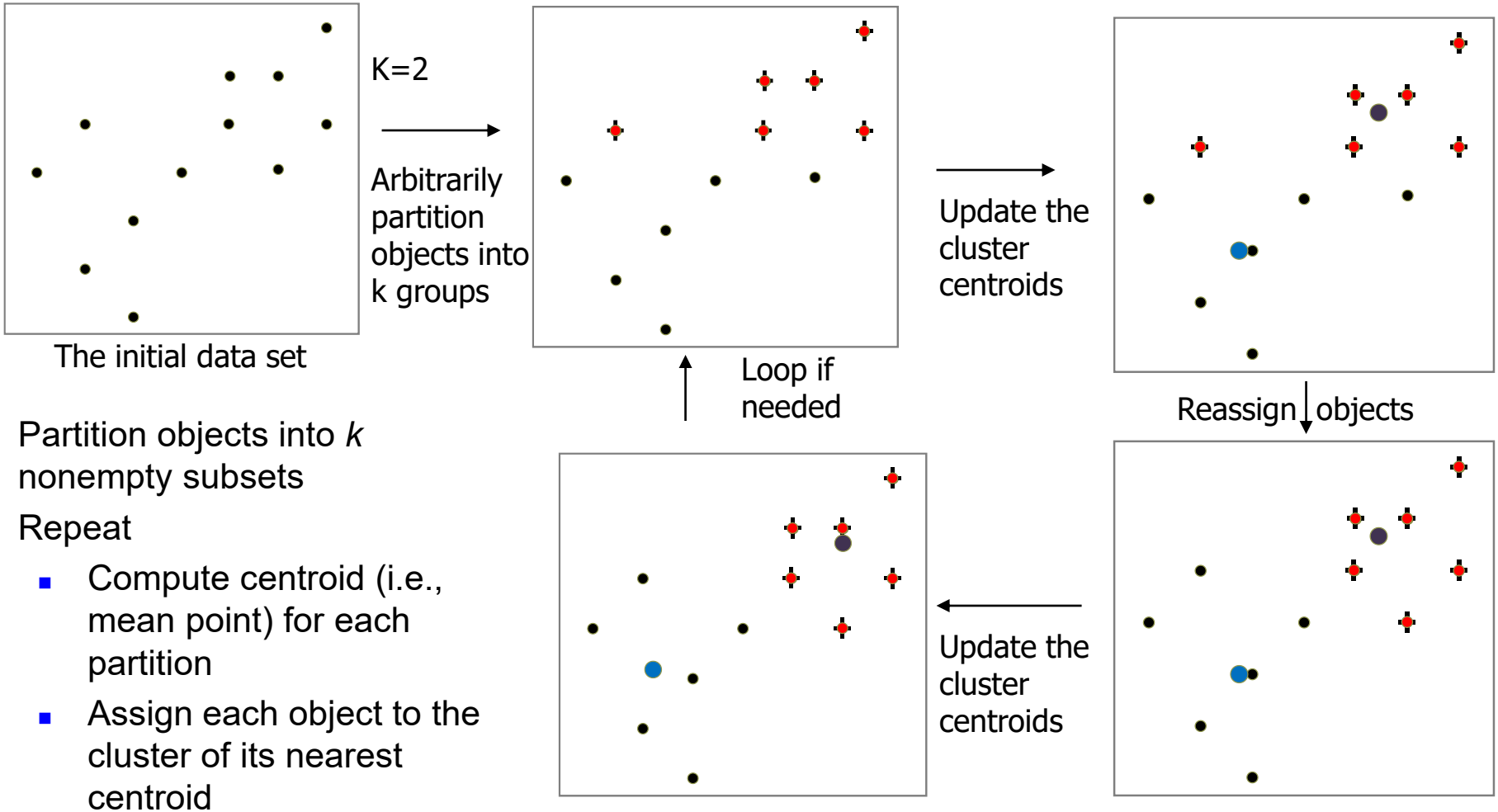
Ryan Benton

August 31, 2018

The *K-Means* Clustering Method

- Given *k*, the *k-means* algorithm is implemented in four steps:
  - Partition objects into *k* nonempty subsets
  - Compute seed points as the centroids of the clusters of the current partitioning (the centroid is the center, i.e., *mean point*, of the cluster)
  - Assign each object to the cluster with the nearest seed point
  - Go back to Step 2, stop when the assignment does not change

An Example of K-Means Clustering



- Partition objects into *k* nonempty subsets
- Repeat
  - Compute centroid (i.e., mean point) for each partition
  - Assign each object to the cluster of its nearest centroid
- Until no change

High-Level

- Select and clone *K* clusters randomly (no duplicates)
- NumIter = 0; Changed = TRUE;
- While (NumIter < MAXITER) AND (Changed = TRUE)
  - For each data point
    - Calculate Distance to Each Cluster
    - Assign point to closest cluster
  - Update Clusters(Clusters, Points, Changed)

High-Level

- CalculateDistance(Acluster (aka i), Apoint (aka j))
  - NOTE: each point (or cluster) has p features

$$d(i,j) = |x_{i_1} - x_{j_1}| + |x_{i_2} - x_{j_2}| + \ldots + |x_{i_p} - x_{j_p}|$$

- ClosestCluster
  - Take minimum of the distances.

High-Level

- UpdateCluster(Cluster, PointsBelongingToCluster)
  - For (j = 0; j < p; j++)
    - FeatureTotal = $\sum_{i=1}^{n} f_{ij}$
    - $\text{Cluster}_j = \frac{\text{FeatureTotal}}{N}$
    - NOTE:
      - N == number of points
      - p -- number of features
      - f_{ij} -- the jth feature value of the ith point