K-means clustering

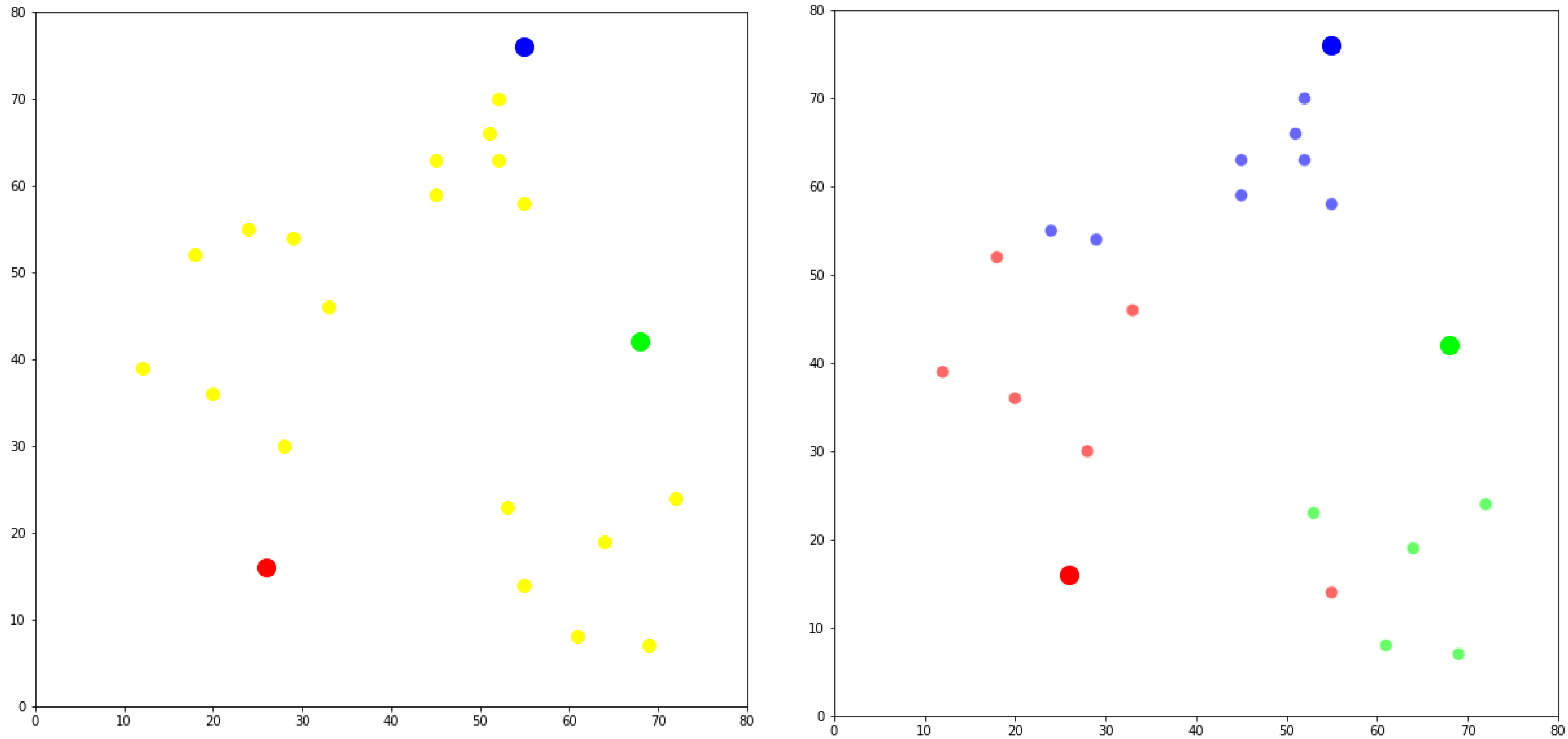
- Clustering : the process of partitioning a group of data points into a small number of clusters e.g. clustering movies by genre
- The k-means clustering algorithm classifies ‘n’ points into ‘k’ clusters by assigning each point to the cluster whose average value on a set of p variables is nearest to it
- Behavioral segmentation, Inventory categorization, Detecting anomalies etc...

K-means clustering

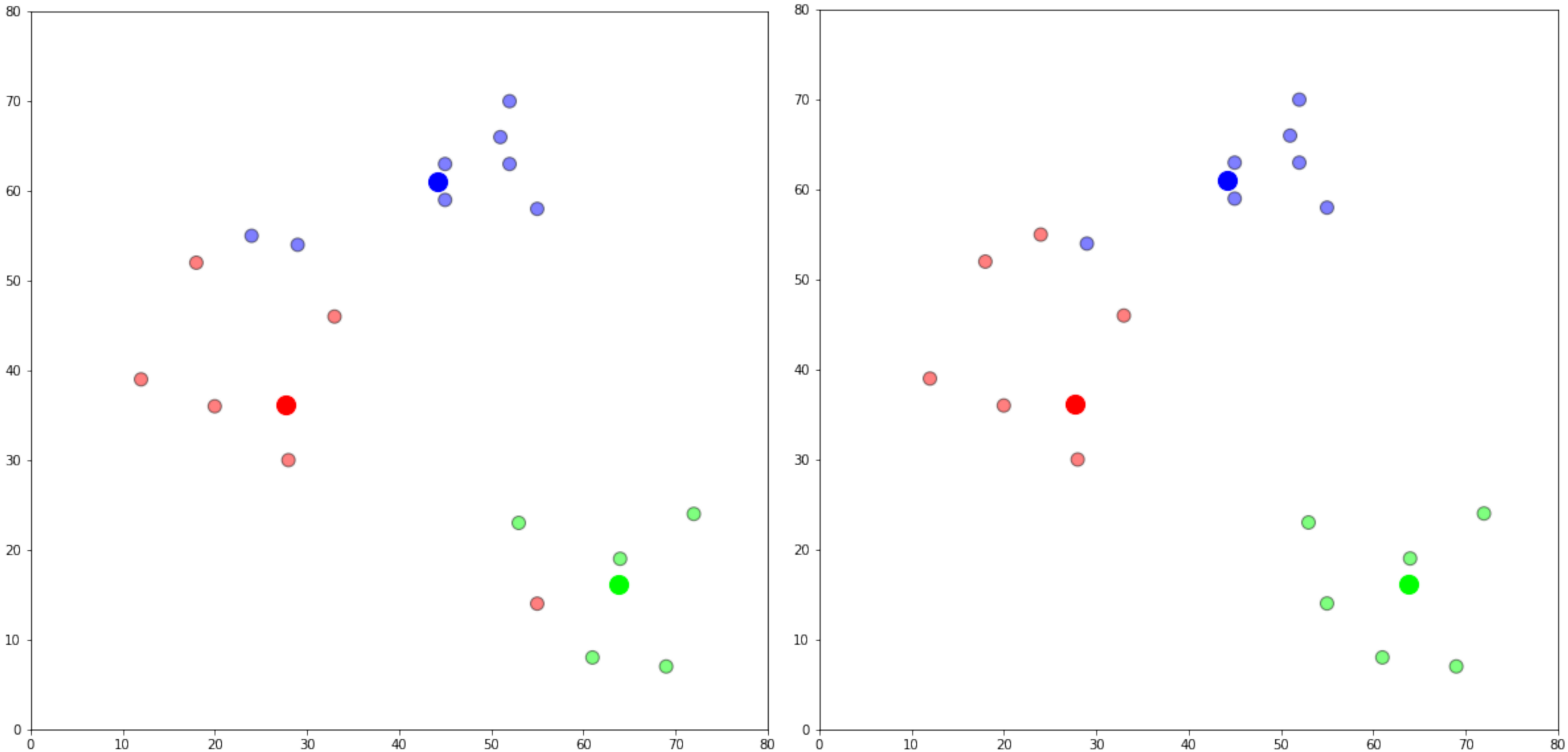
K-means Algorithm

1. Randomly select ‘k’ cluster centers
2. Calculate the distance between each data point and cluster centers
3. Assign the data point to the cluster center whose distance from the cluster center is minimum of all the cluster centers
4. Recalculate the new cluster
5. Recalculate the distance between each data point and new obtained cluster center
6. Repeat until no data point was reassigned

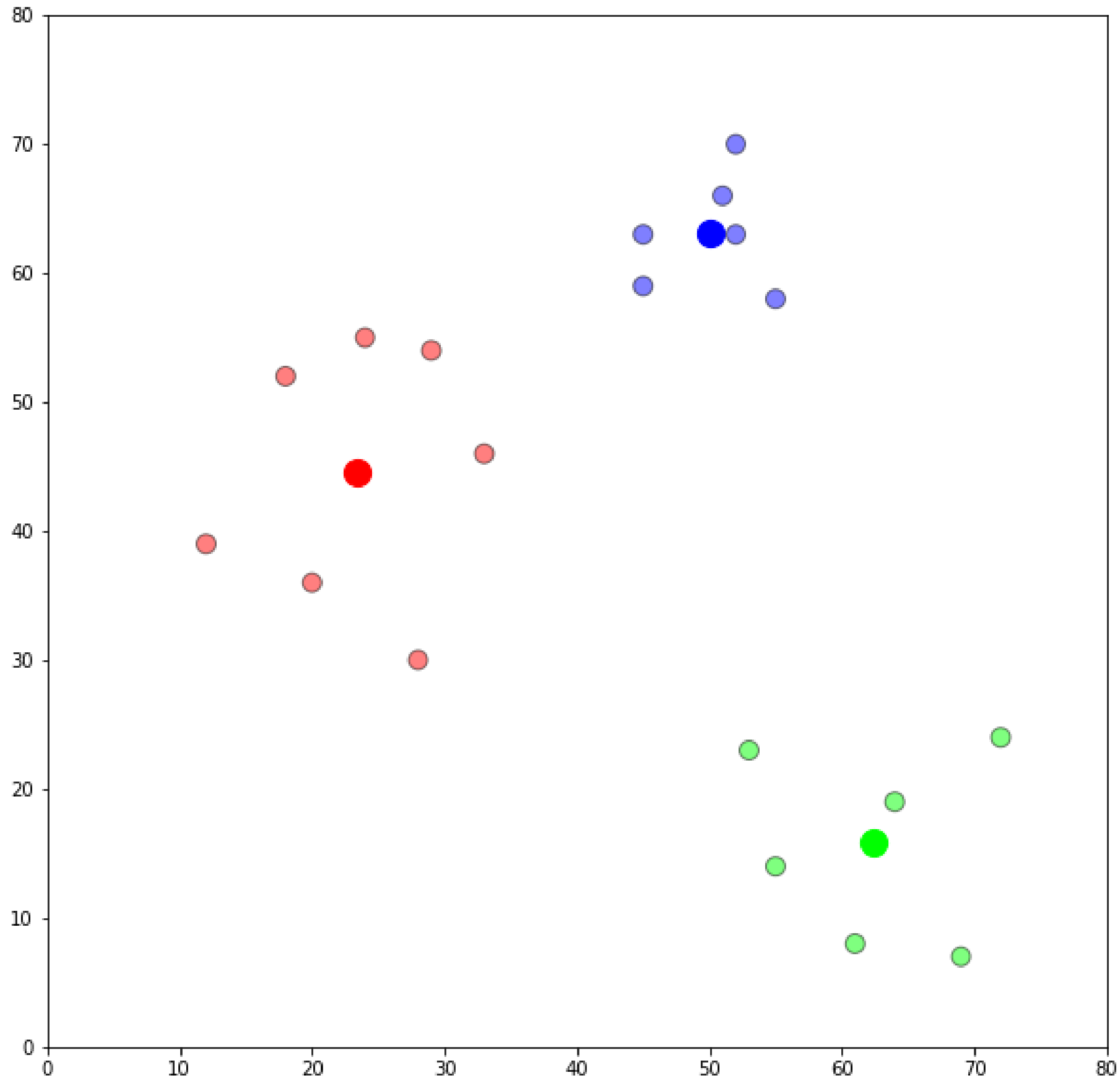
K-means clustering



K-means clustering



K-means clustering



K-means clustering

- Let’s concentrate on computing the distance between the cluster center and the points
- Let’s assume that distance metric is euclidean distance

$$distance = \sqrt{\sum_{i=1}^{D} (K_i - P_i)^2}$$

- We can ignore square root
- How do we implement this using a GPU

K-means clustering

- We can assign each point to each thread.
- For a given point each thread would compute the distance between K centers
- Can we use matrix multiplication to solve this problem?

K-means clustering

$$distance = \sum_{i=1}^{D} (K_i - P_i)^2$$

- Lets expand the above formula (we ignored sqrt)

$$distance = \sum_{i=1}^{D} K_i^2 + P_i^2 - 2K_i * P_i$$

- How can we compute the first term $\sum_{i=1}^{D} K_i^2$

K-means clustering

- How can we compute the first term? $\sum_{i=1}^{D} K_i^2$
- This is a simple dot product
- Does the first term change across iterations?
- We can compute the second term similarly $\sum_{i=1}^{D} P_i^2$
- Does the second term change across iterations?

K-means clustering

- What about the third term $\sum_{i=1}^{D} -2K_i * P_i$

- Let K_i^j represent the the i^{th} dimension of j^{th} point. The notation for P is similar

$$P = \begin{bmatrix} p_1^1 & p_2^1 & p_3^1 & \dots & p_D^1 \\ p_1^2 & p_2^2 & p_3^2 & \dots & p_D^2 \\ \dots \\ p_1^N & p_2^N & p_3^N & \dots & p_D^N \end{bmatrix} \quad C^T = \begin{bmatrix} c_1^1 & c_2^1 & c_3^1 & \dots & c_K^1 \\ c_1^2 & c_2^2 & c_3^2 & \dots & c_K^2 \\ \dots \\ c_1^D & c_2^D & c_3^D & \dots & c_K^D \end{bmatrix}$$

- Compute $P*C^T$

K-means clustering

- Now for each row take the minimum of $P*C^T$ to find the cluster id
- After the above step we know the new cluster id's of each point
- How do we compute the cluster center? (does this problem look similar to what we done before)