Clustering

What is Clustering

- Group observations together so that the elements in one group will be
  - Similar to one another
  - Different from elements in other groups

- Unsupervised Learning
  - The clustering rules are coming from the data, not from external specifications

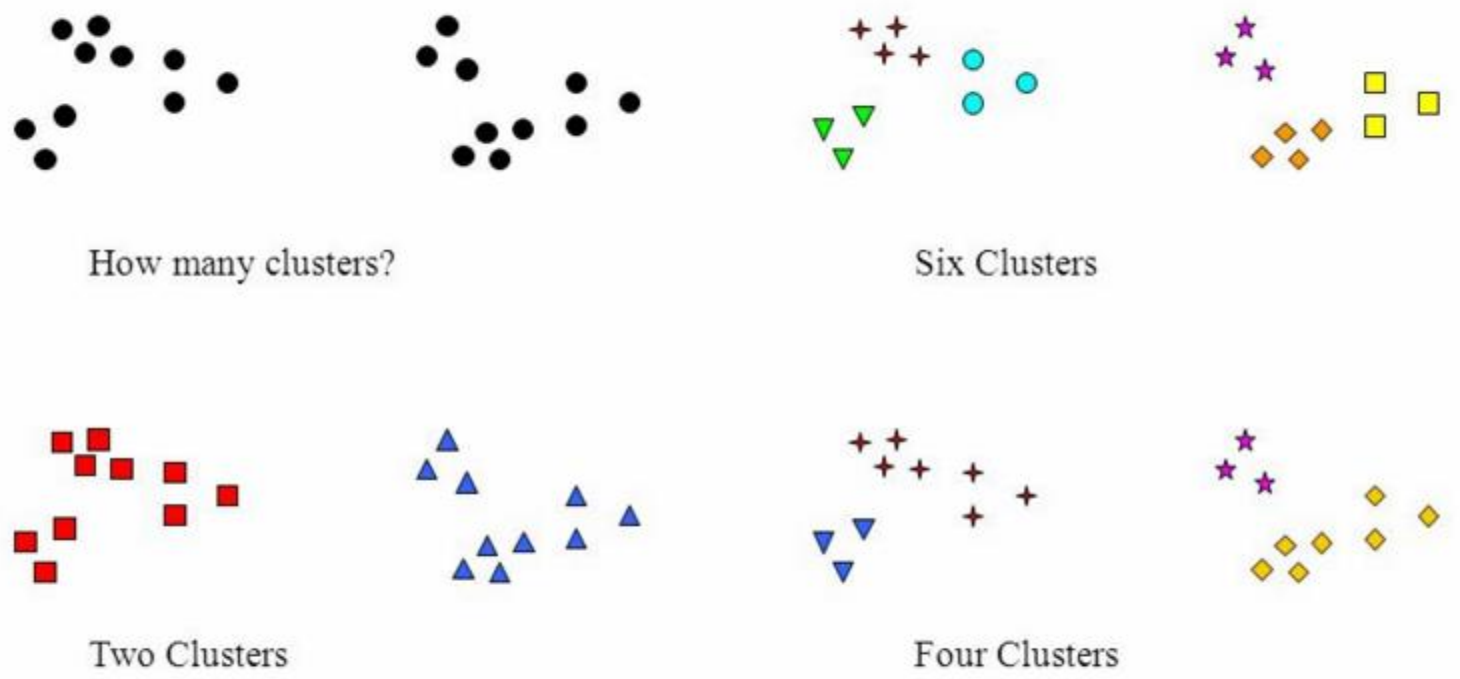
Clustering Application

- Discover which stocks share similar market fluctuations
  - Samsung and PC accessary manufacturers (same direction)
  - Apple and Samsung (maybe opposite direction)

- Group customers according to their attributes
  - Targeted Marketing (send Amazon Prime advertisement to frequent buyers)
  - Finding "Good" customers (never claim refund, return, and care less about discounts)

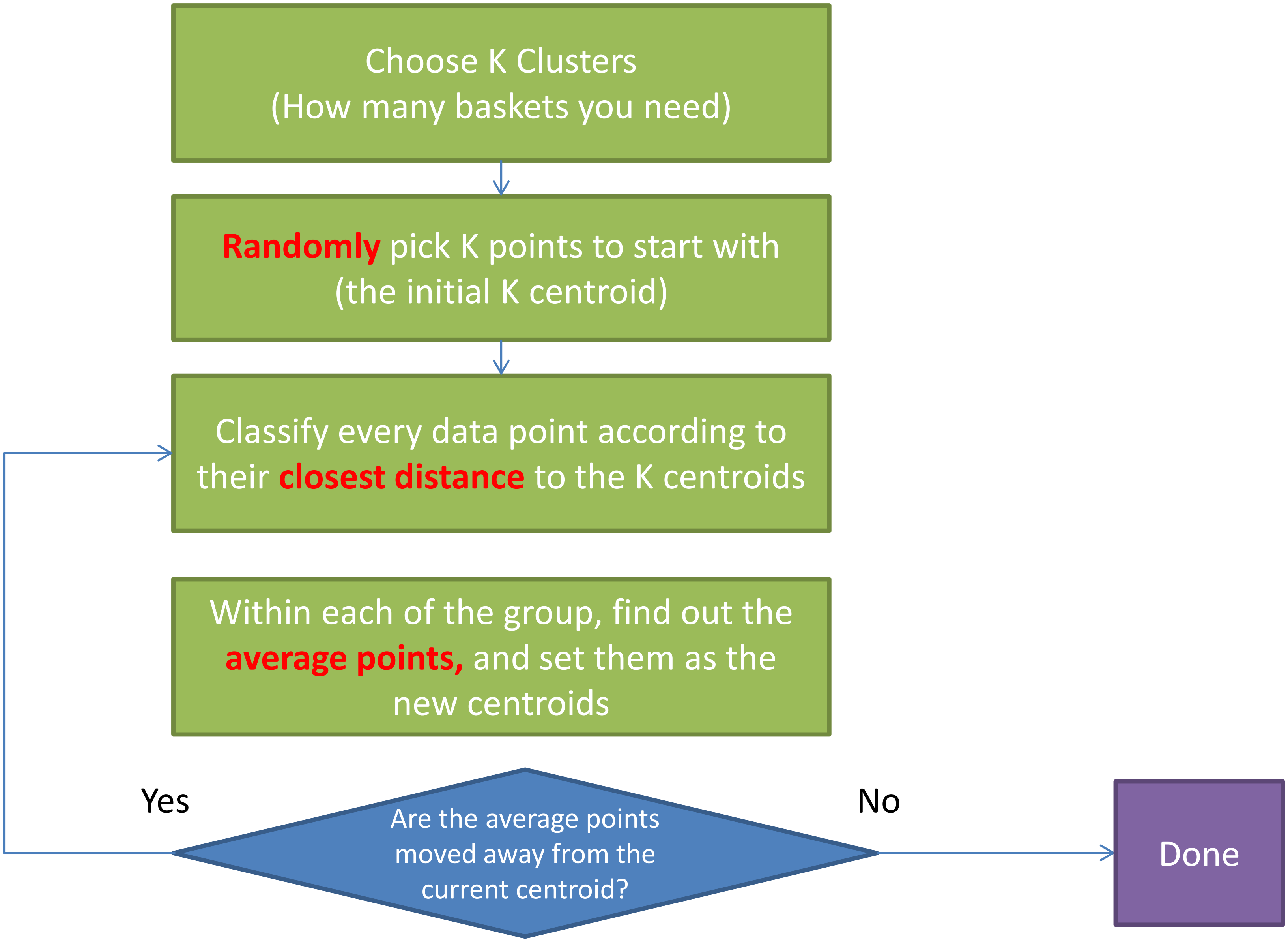
- Group borrowers according to their repayment behavior
  - Predict who is going to default

Clustering Can be Ambiguous

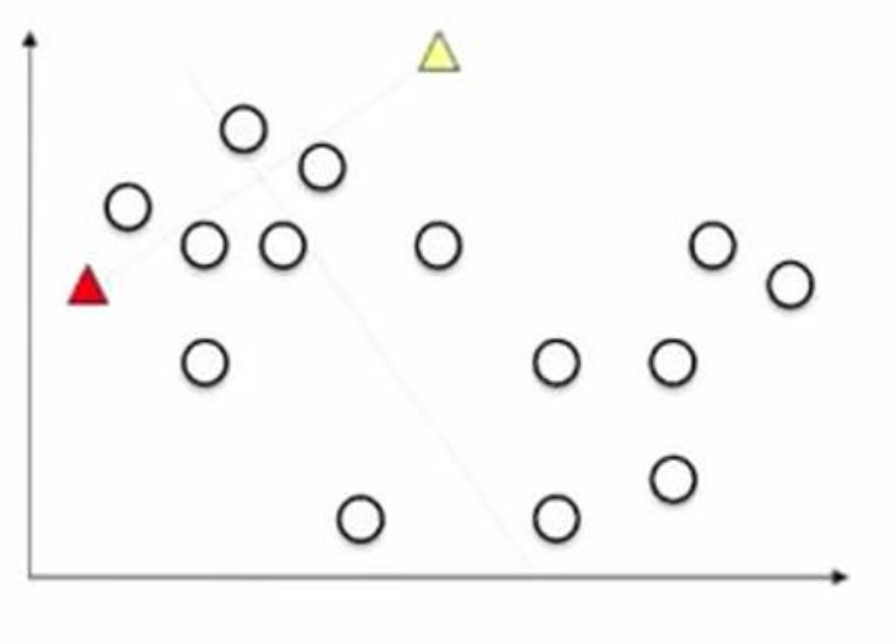


Based on different attributes, we got different clusters

K – Means Clustering

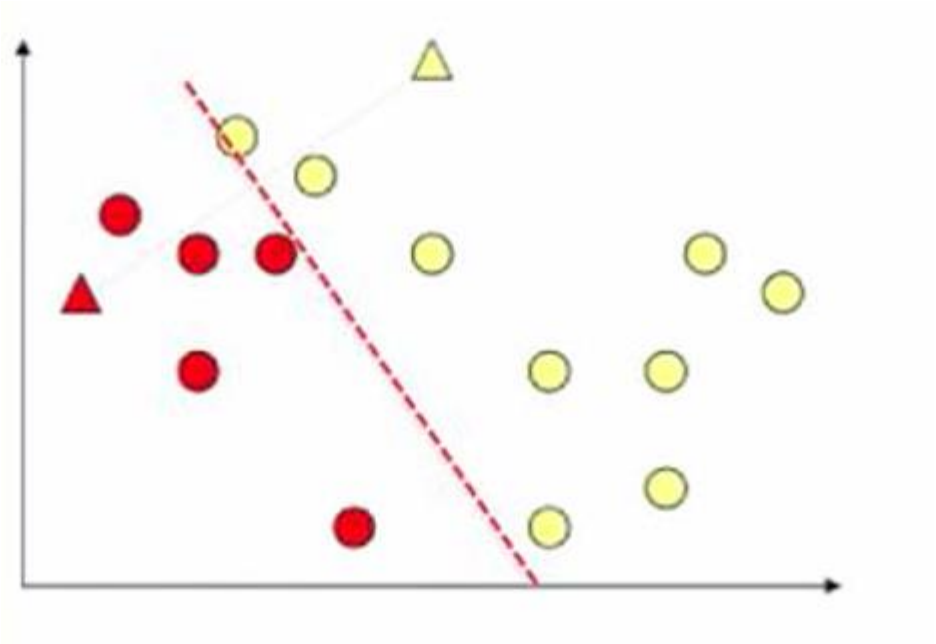


K – Means Clustering



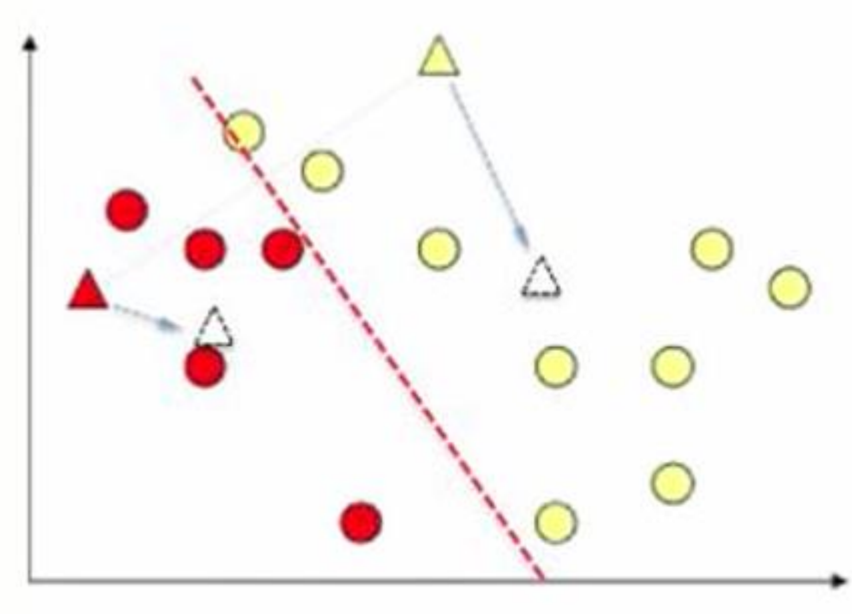
We have these circles, and we try to cluster them
And there are two attributes: x and y, or x1 and x2
We randomly select 2 centroids: red and yellow

K – Means Clustering



- We calculate the distance between each data point and the two temporary centroids.
- If a point is closer to red, it is classified as red. If a point is closer to yellow, it is classified as yellow.

K – Means Clustering

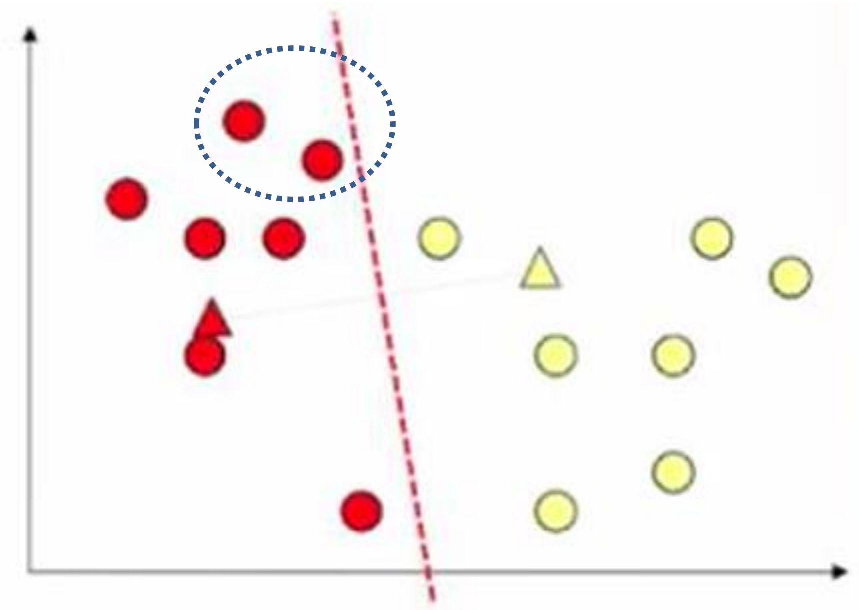


- Within each temporary group, we calculate the central point. For example, the new red centroid is $(\bar{x}_j, \bar{y}_j)$, where $j \in Red$ Like the central gravity point.

- Do the same to the yellow group.

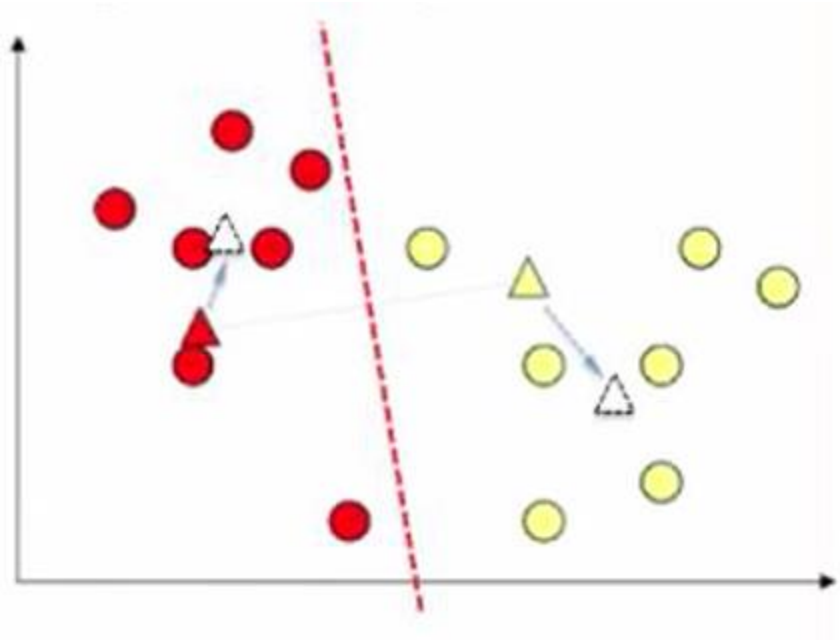
- Because the centroids moved, we have to reclassify each points using the new centroids.

K – Means Clustering



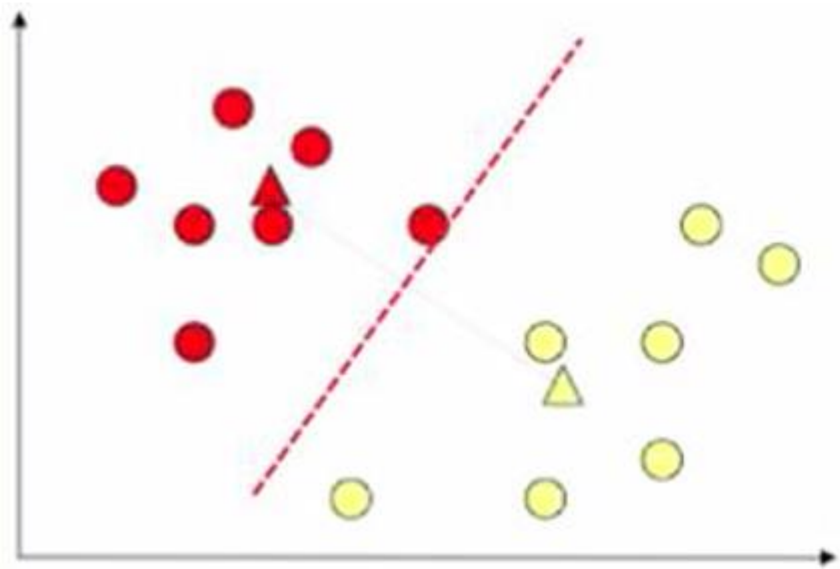
- Here is the new grouping, the circled points are now belong to the red group
- Then we calculate the new centroids based on the updated grouping

K – Means Clustering



- The new centroid moved again.
- So we need to repeat the process
- Until……

K – Means Clustering

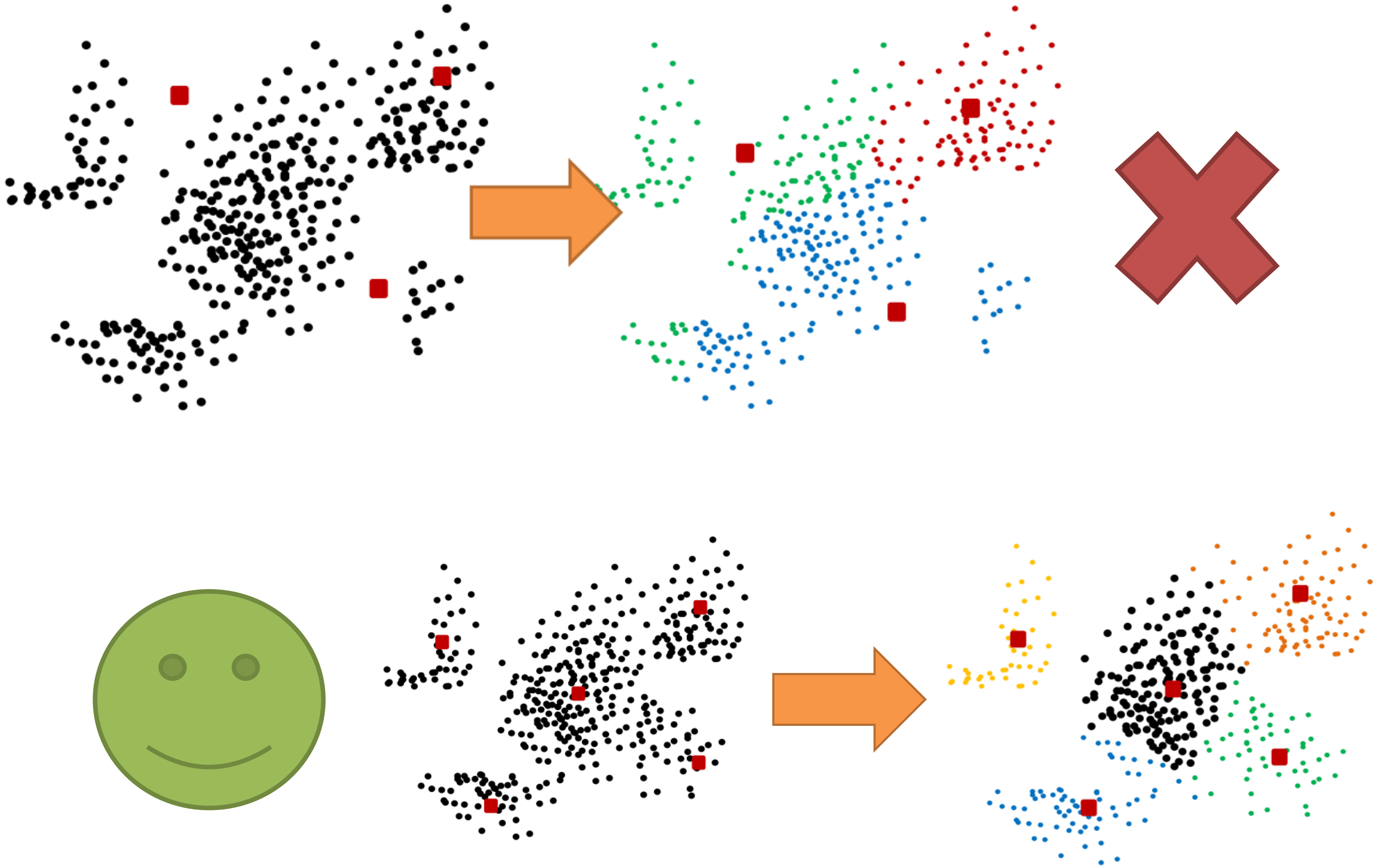


- Until we get to a point where the “central gravity point” does not move at all.
- In most cases, the centroids will converge very quickly.

Choosing the Initial Centroid

- This is important
  - Choosing the right number (how many clusters you need)
  - Choosing the right location (where you put your initial centroids)
- If this is not done properly
  - Mathematically no problem, the algorithm will still work
  - But the clustering will not make sense

Choosing the Initial Centroid



Evaluating K-Mean Clusters

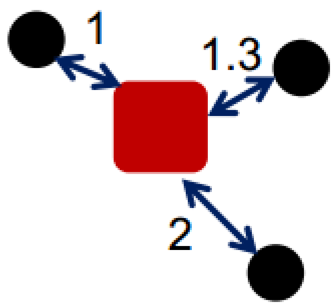
- Sum of Squared errors (SSE)

- Individual Cluster SSE = $\sum_{x \in C_i} dist^2\,(C_i, x)$
  - How close each of the points in cluster i to the center
  - Lower individual SSE – Better individual cluster

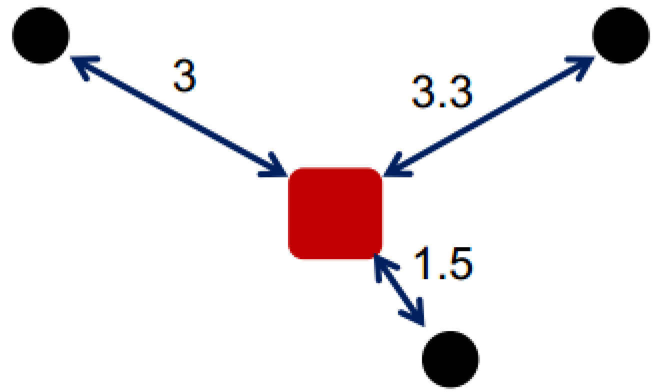
- Total SSE = $\sum_{i=1}^{k} \sum_{x \in C_i} dist^2\,(C_i, x)$ = sum of every individual SSE
  - Lower Total SSE – a better set of clusters

Evaluating K-Mean Clusters

Cluster 1



Cluster 2



$SSE_1 = 1^2 + 1.3^2 + 2^2$
$= 1 + 1.69 + 4 = 6.69$

$SSE_2 = 3^2 + 3.3^2 + 1.5^2$
$= 9 + 10.89 + 2.25 = 22.14$

For this particular clustering result:
Individual SSE: SSE1=6.69, SSE2=22.14
Total SSE=SSE1 + SSE2= xxx
If there is another clustering result that has lower Total SSE, maybe we need to pick it.

How to Choose the Best Initial Centroid

- There is no single, best way to do it
- Some suggestions
    - Select more centroids to start with, and then choose the ones that are farthest apart.
    - Pre and post processing the data

Pre/Post-Process the Data

Pre-Processing

- Normalize the data

- Remove outliers
  - Outliers don't represent the population, remove them

Post-Processing

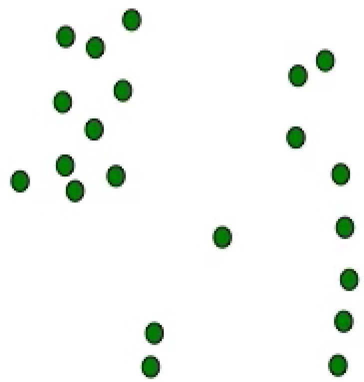
  - Remove small clusters
  - Split loose clusters
  - Merge cluster that are too close
  - So, pretty subjective
  - Until it makes sense
  - If the clusters never make sense, the data may just not be well-suited for clustering

How Many Clusters Are Appropriate

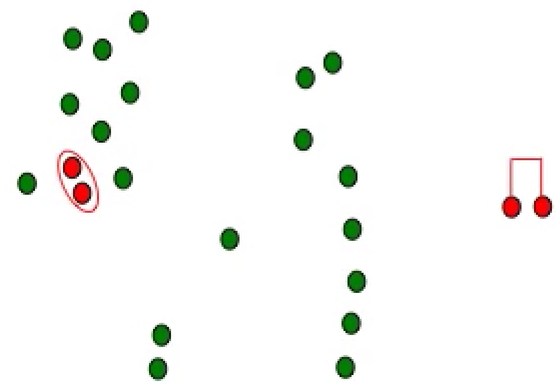
We may use "Hierarchical Clustering" to determine

- First, define a distance (Euclidean Distance etc.)
- Initialize: treat each data point as a cluster
- Compute distance between all clusters
- Merge the closest two clusters
- Repeat the previous steps

How Many Clusters Are Appropriate

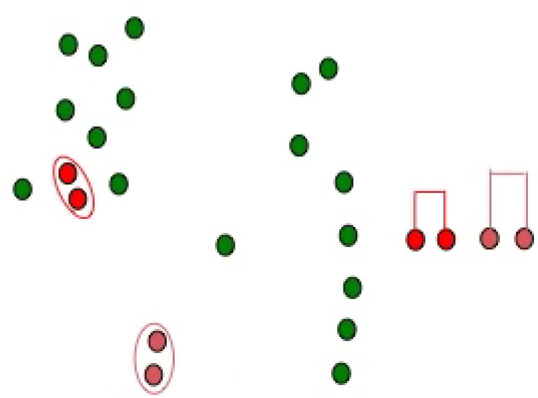


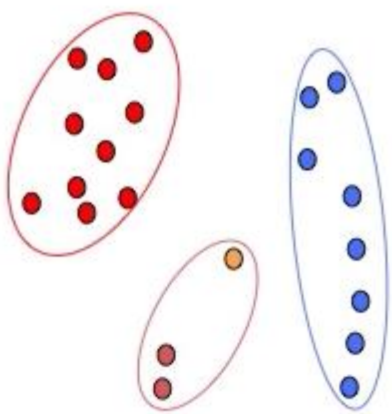
Treat each point as a cluster

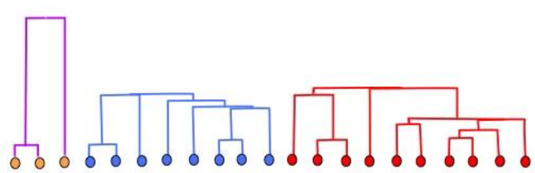


Find the two closest clusters and merge them

Recalculate distance and merge the next closest clusters



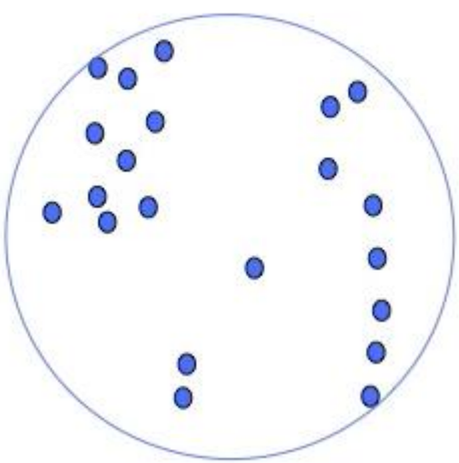


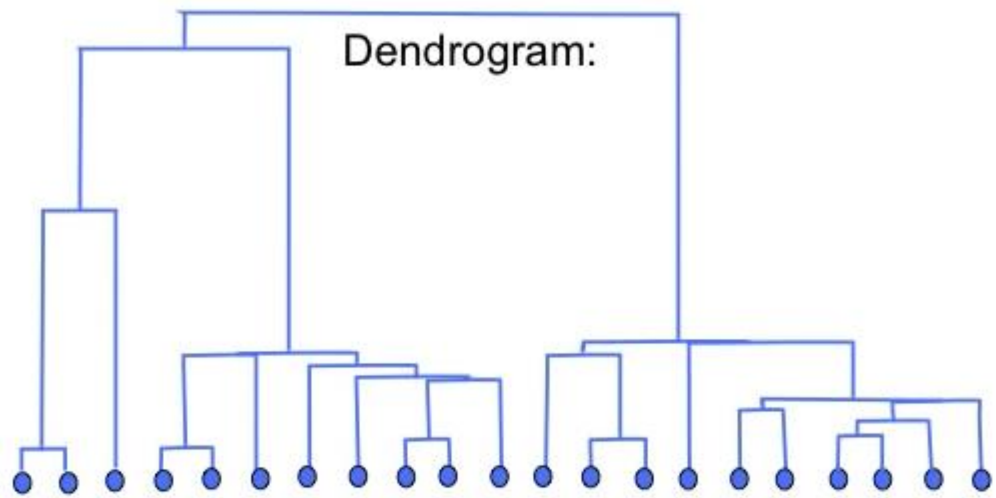


Eventually we end up with only a few clusters

How Many Clusters Are Appropriate

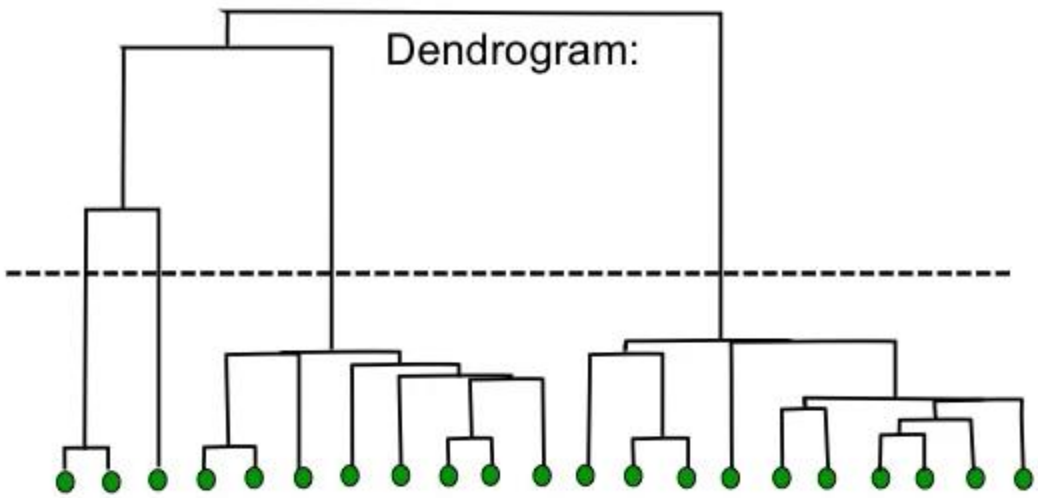
- If we keep merging nonstop, we end up with only one cluster
- Where every data point is in this single cluster
- So we have to stop somewhere





How Many Clusters Are Appropriate

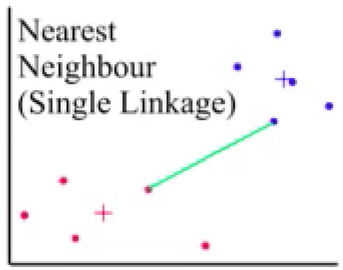
- We can choose a dissimilarity threshold.
- When two cluster is dissimilar to some extent, we stop merging



How Many Clusters Are Appropriate

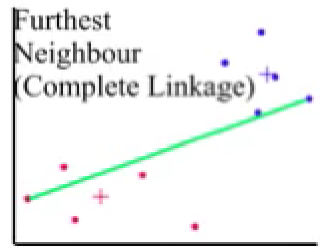
- Some dissimilarity measures

$$D_{min}(C_i, C_j) = min_{x \in C_i, y \in C_j} \|x - y\|^2$$



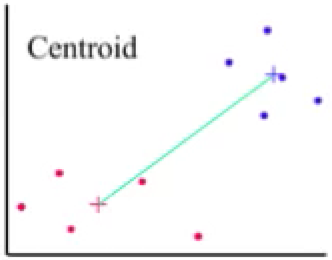
produces minimal spanning tree.

$$D_{max}(C_i, C_j) = max_{x \in C_i, y \in C_j} \|x - y\|^2$$



avoids elongated clusters.

$$D_{means}(C_i, C_j) = \left\|\mu_i - \mu_j\right\|^2$$



$$D_{avg}(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{x \in C_i, y \in C_j} \|x - y\|^2$$