Using K-means to Segment Customers

Professor Song Yao
Olin Business School

Customer Analytics

We have learned RFM

RFM Basics

- Pick up to three features of customers (R, F, M)
- Rank and divide customers into *K* segments based on their R
- Repeat the rank and divide procedure for F and M, respectively
- Cluster customers into more granular segments of the combination of R, F, M segments
- Test and target, or design different initiatives for various segments

Limitations of RFM

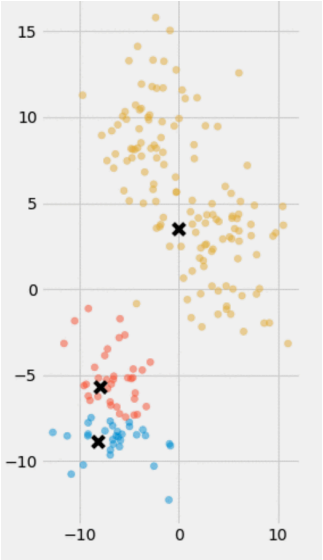
- Can only consider a handful features
- Implemented in a rule-of-thumb manner. E.g.,
  - Why 5 instead of 10 groups per feature?
  - How to decide the size of each group in a feature (i.e., why equal size)?
  - Consider the interdependence of R, F, M in an ad hoc manner (sequantial RFM).

K-means

- An unsupervised machine learning algorithm
- Clustering data into K distinct groups based on their similarity.
  - This is exactly the purpose of customer segmentation:
    - Within group, customers are as similar as possible
    - Across group, customers are as different as possible

The basic idea of K-means

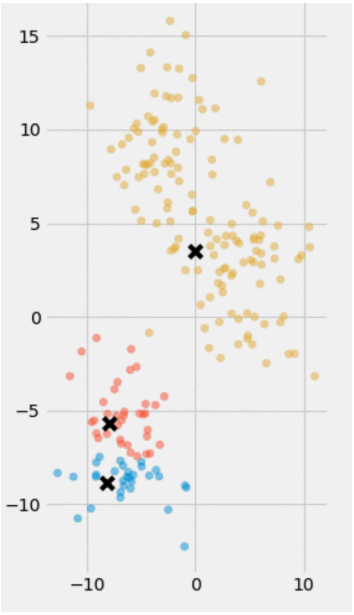
- **Grouping by Similarity**: The goal is to group data points that are similar to each other into clusters based on their features.
- **Centroids as Representatives**: Each cluster is represented by a central point (centroid), which is the mean of all points in the cluster.
- **Assignment**: Data points are assigned to the cluster with the closest centroid, ensuring they belong to the group they are most similar to.



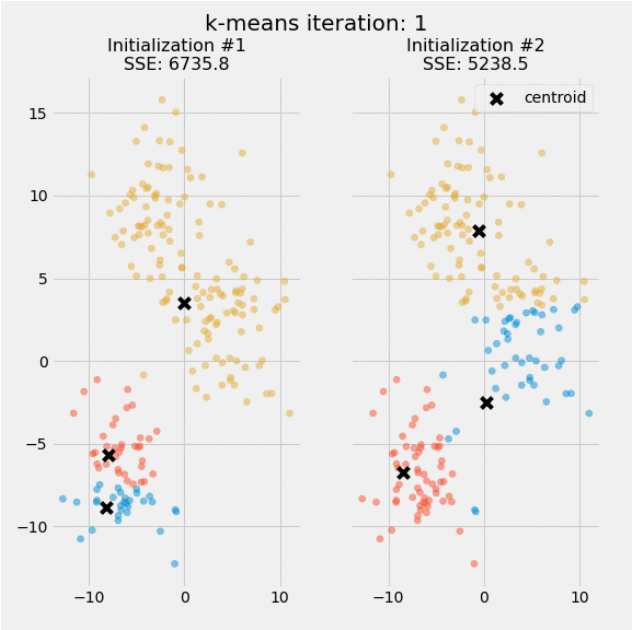
Some math (feel free to forget, but I hope you don't)

$$WCSS = \sum_{i=1}^{k} \sum_{x \in C_i} \| x - \mu_i \|^2$$

- $WCSS$: Within-cluster sum of squares
  - Sometimes also called SSD, sum of squared distances.
- k: Number of clusters
- C_i: Cluster i
- x: A data point in Cluster i
- μ_i: The centroid of Cluster i
- $\| x - \mu_i \|^2$: The squared Euclidean distance between x and μ_i



A toy animation of the K-means algorithm



Credit: https://realpython.com/k-means-clustering-python/

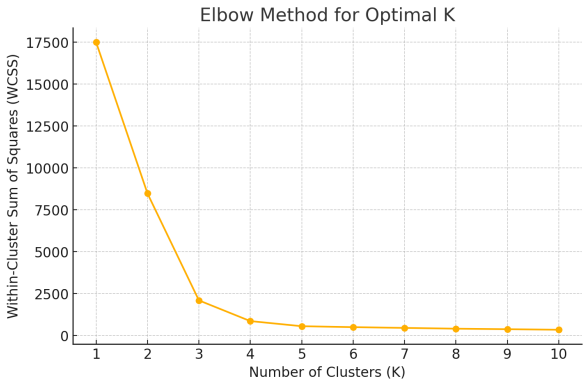
Another animation: https://www.naftaliharris.com/blog/visualizing-k-means-clustering/

Algorithm of K-means

1. **Initialize Centroids**: Choose *K* initial centroids randomly or using specific methods (e.g., K-means++).
2. **Assignment Step**: Assign each data point to the nearest centroid based on Euclidean distance.
3. **Update Step**: Recalculate the centroid of each cluster by taking the mean of all points assigned to that cluster.
4. **Repeat**: Repeat Steps 2 and 3 iteratively until centroids no longer change significantly or a maximum number of iterations is reached.
5. **Output**: Return the final cluster assignments and centroids.

How to decide the number of cluster/segments *K*?

- Elbow method



- Silhouette Score, Gap Statistics, and others

How does K-means help us target?

- The idea is similar to RFM
  - Segment using K-means
  - Test round to evaluate the response rate of each segment
  - Decide the break-even response rate
  - Based on the comparison between each segment's response rate and the break-even, decide which segment to target

Let's try it on the BBB data

- Using only R, F, M so we can compare with RFM method
- Code can be found on Canvas