Lecture04: k-Means Cluster Analysis

See Chapter 20 in Boehmke *et al.* and Chapter 10 in Gareth *et al.*

- In cluster analysis no prior knowledge of the grouping labels y of the individual objects $\{1,2,\dots,n\}$ exists. Thus, it belongs to the class of **unsupervised** machine learning techniques.

Objectives

- Group observations together into clusters with regards to features describing each observation
- The clusters are supposed to be internally as homogeneous as possible with regards to features in their cluster (internal compactness).
- The clusters are supposed to be as heterogenous as possible between each other cluster (maximum separation).

Example: Potential Applications (Brett Lantz)

- Segmenting customers into groups with similar demographics or buying patterns for targeted marking campaigns.
- Detecting anomalous behavior, such as unauthorized network intrusions, by identifying patterns of use falling outside known clusters
- Simplifying extremely large datasets by grouping similar feature values into smaller number of homogeneous categories.

Challenges: Strength and Weaknesses

- Identifying features that best describe the objects and their clusters.

- Interpretation of the “meaning” of each cluster.

Strengths	Weaknesses
• k-Means uses a simple geometric algorithm • k-Means is a highly flexible method that can easily be adjusted to overcome some of its shortcomings. • k-Means has a proven tract record of performing well for many empirical datasets.	• Not as sophisticated as several modern clustering algorithms. • Its use of random chance leads to slightly different outcomes from analysis to analysis. • It can get stuck in at local optima. • Requires a reasonable guess into how many distinct clusters K a dataset breaks up. • Not ideal for non-compact clusters or clusters with widely varying densities.

- Example: Unknown number of clusters K:

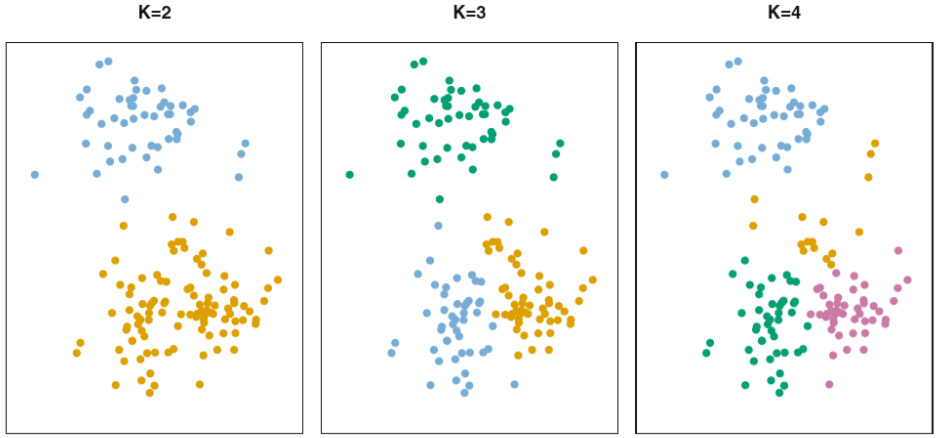

FIGURE 10.5. *A simulated data set with 150 observations in two-dimensional space. Panels show the results of applying K-means clustering with different values of K, the number of clusters. The color of each observation indicates the cluster to which it was assigned using the K-means clustering algorithm. Note that there is no ordering of the clusters, so the cluster coloring is arbitrary. These cluster labels were not used in clustering; instead, they are the outputs of the clustering procedure.*

Mathematical Formulation of Clusters

- Break a set of n objects up into K cluster $C_k$so that
  1. $C_1 \cup C_2 \cup \ldots \cup C_K = \{1,2,\ldots,n\}$: exhaustive classification of all objects
  2. $C_k \neq \{\ \}$ for all $k \in \{1,2,\ldots,K\}$: no unessential clusters
  3. $C_k \cap C_{k'} = \{\ \}$ for all $k \neq k'$: disjunct, that is, on object belongs to more than one cluster.
- The disjunct property implies, that the class membership is deterministic and not probabilistic.

k-Means as Optimization Problem

- The aim is to minimize ***objective function*** of the ***internal heterogeneity*** $W(C_k)$ of a set of clusters:

$$\min_{C_1,C_2,\ldots,C_K} \sum_{k=1}^{K} W(C_k)$$

- The intra class heterogeneity can be defined as the features distances within a class:

$$W(C_k) = \frac{1}{|C_k|} \sum_{i,j \in C_k} \sum_{p=1}^{P} \left(x_{ip} - x_{jp}\right)^2$$

$$= 2 \cdot \sum_{i \in C_k} \sum_{p=1}^{P} \left(x_{ip} - \bar{x}_{kp}\right)^2$$

- In total there are $S(n,K) = \frac{1}{K!} \cdot \sum_{k=1}^{K} (-1)^{K-k} \cdot \binom{K}{k} \cdot k^N$ different possibilities to assign n objects into K clusters. E.g., $S(10,4) = 34{,}105$ or $S(19,4) \approx 10^{10}$.
- Consequently, a total enumeration of all possible cluster configuration is impossible and a heuristic search algorithm must be applied.

k-Means Algorithm

- The classification algorithm applies a divide-and-conquer approach.
- It consists of several iterative steps, which guarantee a decreasing objective function $\sum_{k=1}^{K} W(C_k)$:
    1. Randomly assign a number, from 1 to K, to each observation. These serve as initial cluster assignment.
    2. Iterate until the cluster assignment stop changing:
        a) For each of the K cluster assignments calculate the vector of feature centroids.
        b) Re-assign each object to the cluster k whose centroid is the closest (usually this is the squared Euclidean distance)

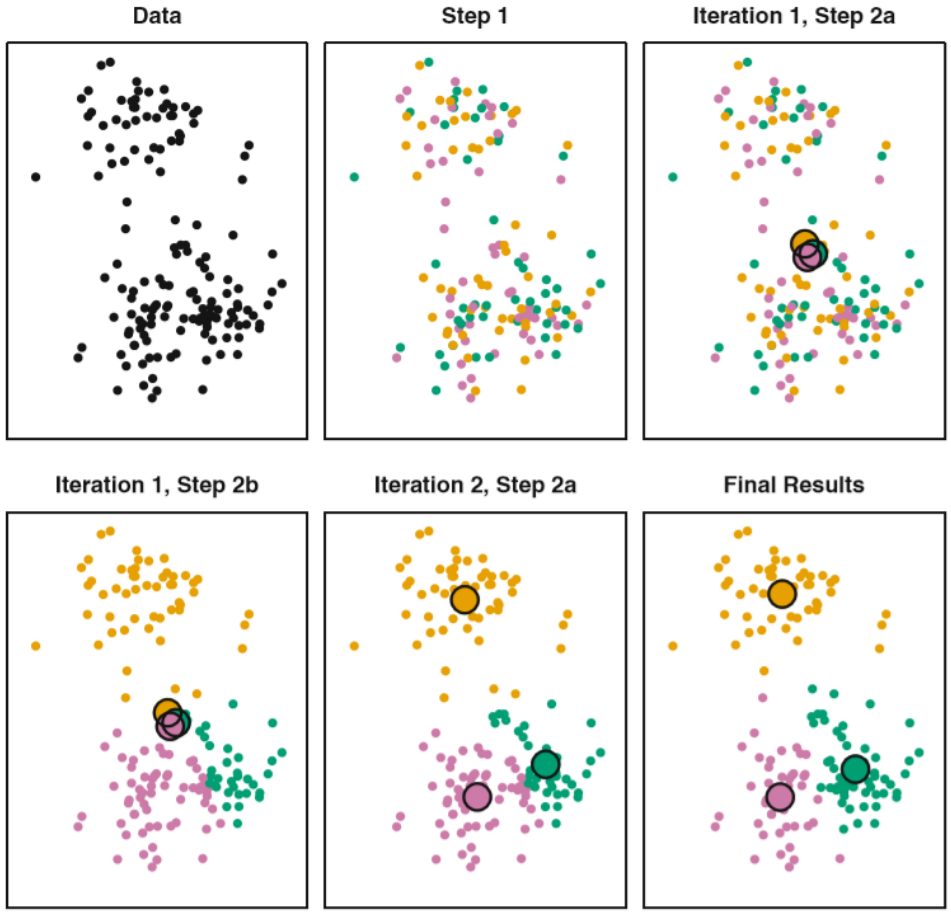

FIGURE 10.6. *The progress of the K-means algorithm on the example of Figure 10.5 with K=3.* Top left: *the observations are shown.* Top center: *in Step 1 of the algorithm, each observation is randomly assigned to a cluster.* Top right: *in Step 2(a), the cluster centroids are computed. These are shown as large colored disks. Initially the centroids are almost completely overlapping because the initial cluster assignments were chosen at random.* Bottom left: *in Step 2(b), each observation is assigned to the nearest centroid.* Bottom center: *Step 2(a) is once again performed, leading to new cluster centroids.* Bottom right: *the results obtained after ten iterations.*

- Because the initial cluster assignment is random, different starting values may lead to different outcomes. One selects that outcome with the smallest sum of the internal cluster heterogeneity.
- This approach cannot be used to determine the number of clusters, because for $K = n$ the internal heterogeneity is zero.
- The possible number of clusters can be determined at that K-value, for which the objective function drops substantially.
- A ***Voronoi polygon*** around each cluster centroid determines the assignment rule of individual objects.
- The interpretation of the clusters is usually done in terms of the cluster centroids, which is used to describe the objects within a cluster.

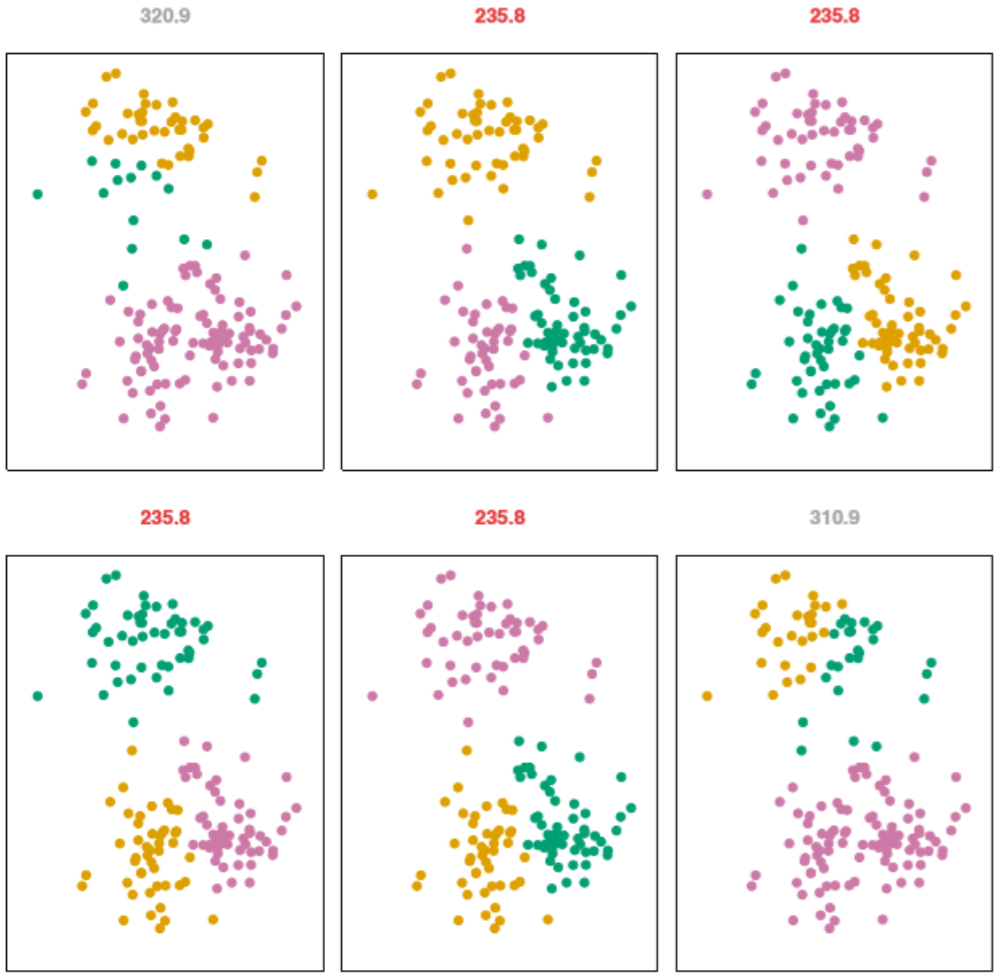

FIGURE 10.7. *K-means clustering performed six times on the data from Figure 10.5 with $K = 3$, each time with a different random assignment of the observations in Step 1 of the K-means algorithm. Above each plot is the value of the objective (10.11). Three different local optima were obtained, one of which resulted in a smaller value of the objective and provides better separation between the clusters. Those labeled in red all achieved the same best solution, with an objective value of 235.8.*