STA 141C: Big Data & High Performance Statistical Computing **Spring 2025**

k-Means & Hierarchical Clustering

1 Continuing forward with k-means clustering

We proceed with our study of k-means clustering with the following general theorem.

Theorem 1.1. *k-means clustering aims to minimize the following objective, with K known a priori:*

$$\underset{C_1,\ldots,C_k}{minimize} \sum_{k=1}^{K} WCV(C_k)$$

where we will define $WCV(C_k)$ as the within-cluster variation for a cluster C_k. In words, this means that k-means clustering aims to partition the observations into K clusters, each with minimal variation.

There are many ways to define the within-cluster variation $\text{WCV}(C_k)$. One natural method is using the Euclidean distance:

$$\text{WCV}(C_k) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} || \, x_i - x_{i'}||^2 = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \tag{1}$$

The k-means algorithm is a famously discussed algorithm. We outline the procedure below.

1. **Initialization.** Choose initial centroids $\{\mu_1^{(0)}, \ldots, \mu_K^{(0)}\}$, typically by selecting K points at random from the dataset.

2. **Assignment step.** For each observation x_i, assign it to the nearest centroid:

$$c_i^{(t)} \;=\; \underset{k=1,\ldots,K}{\text{argmin}} \, \left\| x_i - \mu_k^{(t)} \right\|^2$$

3. **Update step.** Recompute each centroid as the mean of the points assigned to it:

$$\mu_k^{(t+1)} \;=\; \frac{1}{\left| \{ \, i : c_i^{(t)} = k \} \right|} \sum_{i : c_i^{(t)} = k} x_i, \quad k = 1, \ldots, K.$$

4. **Convergence check.** If the assignments $\{c_i^{(t)}\}$ have not changed (or the decrease in the objective (1) is below a threshold), stop; otherwise set $t \leftarrow t + 1$ and return to Step 2.

We now discuss some properties of the k-means algorithm. Note that objective (1) is *guaranteed* to decrease at each iteration (proof: homework). For a hint, observe that we can re-express our objective as:

$$\frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{i,j} - x_{i'j})^2 = 2 \sum_{i \in C_k} \sum_{j=1}^{p} (x_{ij} - \bar{x}_{kj})^2$$

where we define $\bar{x}_{kj}$ as the mean of the feature j in cluster C_k, or more formally $\bar{x}_{kj} = \frac{1}{|C_k|}\sum_{i \in C_k} x_{ij}$. This objective function is not necessarily convex $\implies k-$ means that it is not guaranteed to achieve the guaranteed global minimum. Furthermore, k-means will achieve different clusterings depending on the random initializations.

2 Hierarchical Clustering

We conclude our study of clustering algorithms with hierarchical clustering. Note in hierarchical clustering is that we do not commit a priori to a particular choice of K. We now provide an explanation of *bottom-up* (also known as *agglomerative*) hierarchical clustering algorithms[1].

These clustering algorithms start from placing each individual data point in their own clusters. Then, at each iteration, such algorithms merge the two least dissimilar clusters (more on this later) until all data is contained in the same cluster. This creates a "hierarchy" as shown below (left). From this hierarchy, we can create a *dendrogram* which shows the sequence of clusters merging (right).

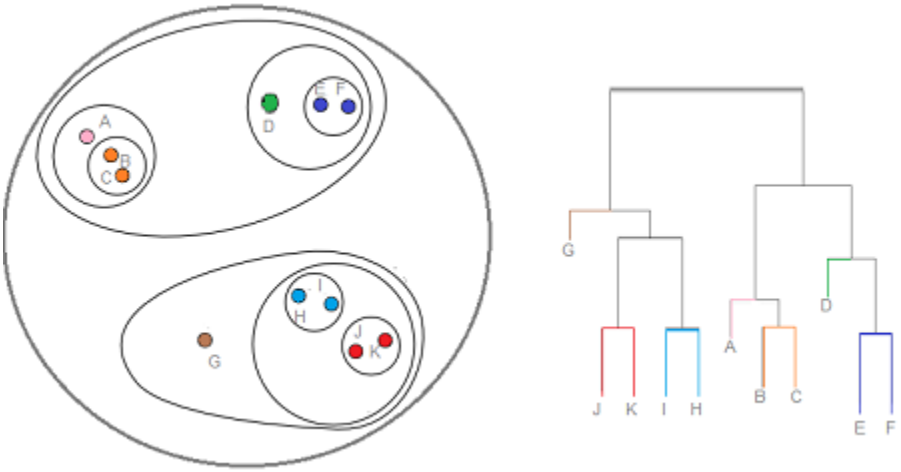

Figure source. Our final step is choosing how to create the different clusters. Using this dendrogram, one method is to create different clusters by choosing different heights to cut the dendrogram into distinct hierarchical groups (i.e. clusters).

There are different ways to measure the dissimilarity between two clusters (which determine which clusters will be merged together.) In practice, we tend to use the complete & average linkage methods, which will measure the dissimilarity[2] between two clusters as the observed maximum & average (respectively) pairwise dissimilarity measure between points across those two clusters.

(Hierarchical) clustering comes with its challenges. To start, there is no standard rule on how to choose how many clusters to have nor on which linkage function. As a precursory warning, note that because all clustering algorithms heavily rely on distance metrics, it is imperative to standardize the features of observations (typically to have a zero-mean and a standard deviation of one.)

[1]For a more visual treatment of these procedures, please see the class slides.

[2]Note that we do not need to use a mathematically rigorous distance function for our dissimilarity measure. While we typically will use the Euclidean distance, a popular alternative is *correlation-based distance* which measures the correlation between the features of two observations.