Cluster Analysis

Prof. Thomas B. Fomby
Department of Economics
Southern Methodist University
Dallas, TX 75275

April 2008
April 2010

Cluster Analysis, sometimes called **data segmentation** or **customer segmentation**, is an **unsupervised learning method**. As you will recall a method is an unsupervised learning method if it doesn't involve prediction or classification. The major purpose of Cluster Analysis is to group together collections of objects (e.g. customers) into "clusters" so that the objects in the clusters are "similar." One reason a company might want to organize its customers into groups is to come to better understand the nature of its customers. Given the delineation of its customers into distinct groups, the company could advertise differently to its distinct groups, send different catalogues to its distinct groups, and the like.

In terms of building prediction and classification models, cluster analysis can help the analyst identify groups of input variables that in turn can lead to different models for each group. This is, of course, assuming that the output relationships vis-à-vis the input variables across the groups are not the same. But then one can always test the "poolability" of the models by either conventional hypothesis tests, when considering econometric models, or accuracy measures across validation and test data partitions when considering machine learning models.

As one will come to understand after working on several clustering projects, clustering is an "Art Form." It must be practiced with care. The more experience you have in doing cluster analysis, the better you become as a practitioner. Before beginning cluster analysis it is often recommended that the data be normalized first. Cluster analysis based on variables with very different scales of measurement can lead to clusters that are not very robust to adding or deleting variables or observations. In this discussion, **we will be focusing on clustering only continuous input variables**. The clustering of mixed data, some continuous and some categorical, is not considered here as it is beyond the scope of this discussion.

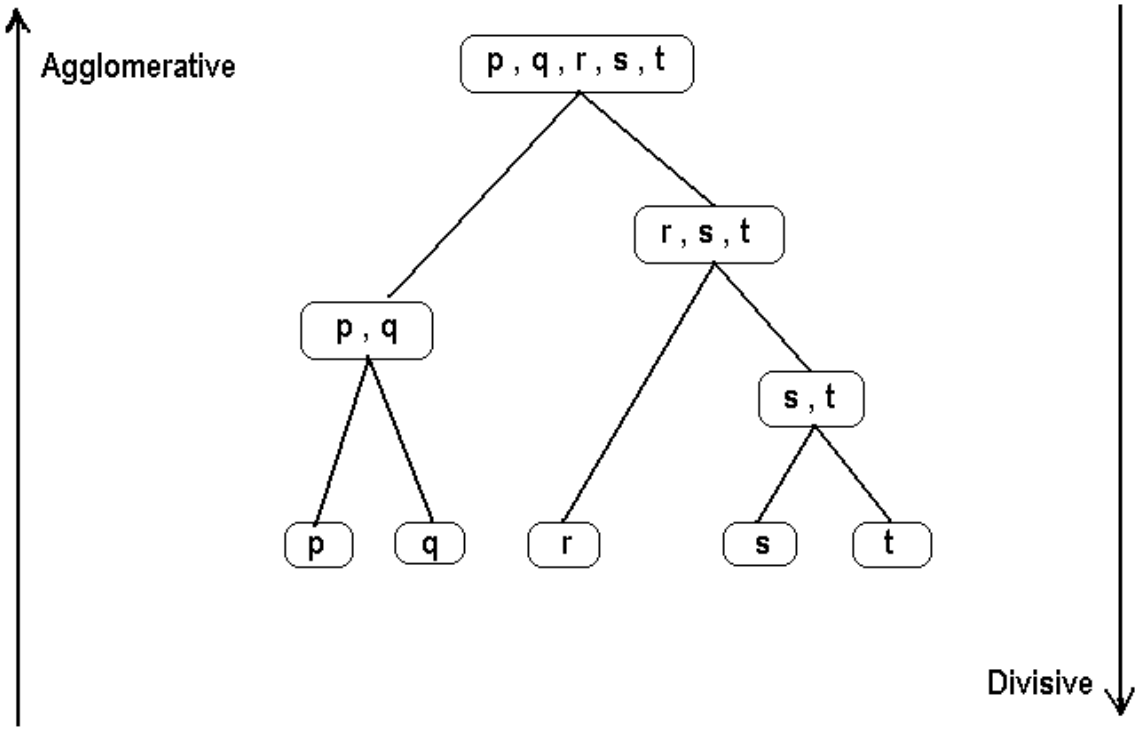
Now let us begin. There are two basic approaches to clustering:
a) Hierarchical Clustering (Agglomerative Clustering discussed here)
b) Non-hierarchical clustering (K-means)

Hierarchical Clustering

With respect to hierarchical clustering, the final clusters chosen are built in a series of steps. If we start with N objects, each being in its own separate cluster, and then combine one of the clusters with another cluster resulting in N – 1 clusters and continue to combine clusters into fewer and few clusters with more and more objects in each cluster, we are engaging in **Agglomerative clustering**. In contrast, if we start with all of the objects being in a single cluster and then remove one of the objects to form a second cluster and then continue to build more and more clusters with fewer and few objects in each cluster until each object is in its own cluster, we are engaging in **Divisive clustering**. The distinction between these two hierarchical methods is represented in the below figure taken from the XLMINER help file.

Figure 1

Hierarchical Clustering:
Agglomerative versus Divisive Methods



The above figure is called a **dendrogram** and represents the fusions or divisions made at each successive stage of the analysis. More formally then, a **dendrogram** is a tree-like diagram that summarizes the process of clustering.

Distance Measures Using in Clustering

In order to build clusters, either agglomeratively or divisively, we need to define the distance between two objects (cases), $(x_{i1}, x_{i2,} \cdots . x_{ip})$ and $(x_{j1}, x_{j2}, \cdots, x_{jp})$ and eventually between clusters. Let us first examine the distance between two objects. If the units of measure of the p variables are quite different, it is suggested that the variables be first normalized by forming z-scores of the variables as in subtracting the sample means from the original variables and dividing the deviations by their respective sample standard deviations. The most often used measure of distance (dissimilarity) between the two cases is the **Euclidean distance** defined by

$$d_{ij} = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \cdots + (x_{ip} - x_{jp})^2} \quad . \qquad (1)$$

Alternatively, a **weighted Euclidean distance** can be used and is defined by

$$d_{ij}^* = \sqrt{w_1(x_{i1} - x_{j1})^2 + w_2(x_{i2} - x_{j2})^2 + \cdots + w_p(x_{ip} - x_{jp})^2} \qquad (2)$$

where the weights $w_1, w_2, \cdots, w_p$ satisfy the properties $w_i \geq 0$ and $\sum_{i=1}^{p} w_i = 1$. For the remaining discussion let us focus on the Euclidean distance measure of distance between objects (cases).

Moving to the discussion of the **distance between clusters** we need to somehow define the distance between the objects in one cluster and the objects in another cluster. **Cluster distances** are usually defined in one of three basic ways: **Single Linkage** (Nearest Neighbor), **Complete Linkage** (Farthest Neighbor), and **Average Group Linkage**. Each of these cluster distance measures are defined in order below:

Single Linkage (Nearest Neighbor)

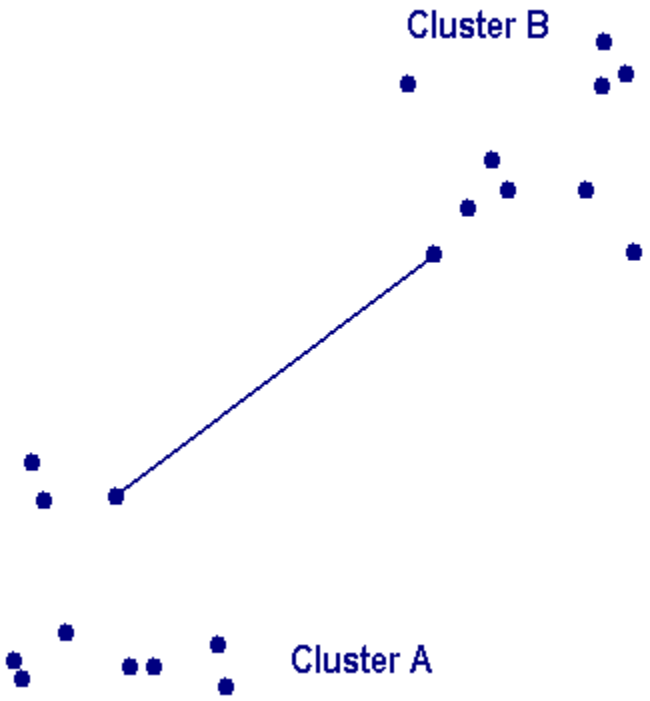
The **Single Linkage distance** between two clusters is defined as the distance between the nearest pair of objects in the two clusters (one object in each cluster). If cluster A is the set of objects $A_1, A_2, \cdots, A_m$ and cluster B is $B_1, B_2, \cdots, B_n$, the Single Linkage distance between clusters A and B is

$D(A, B) = Min\{d_{ij}$: where object A_i is in cluster A and object B_j is in cluster B and d_{ij} is the Euclidean distance between A_i and $B_j\}$

At each stage of hierarchical clustering based on the Single Linkage distance measure, the clusters A and B, for which D(A, B) is minimum, are merged. The Single Linkage distance is represented in the XLMINER Help File figure below:

Figure 2

Single Linkage Distance
Between Clusters



Complete Linkage (Farthest Neighbor)

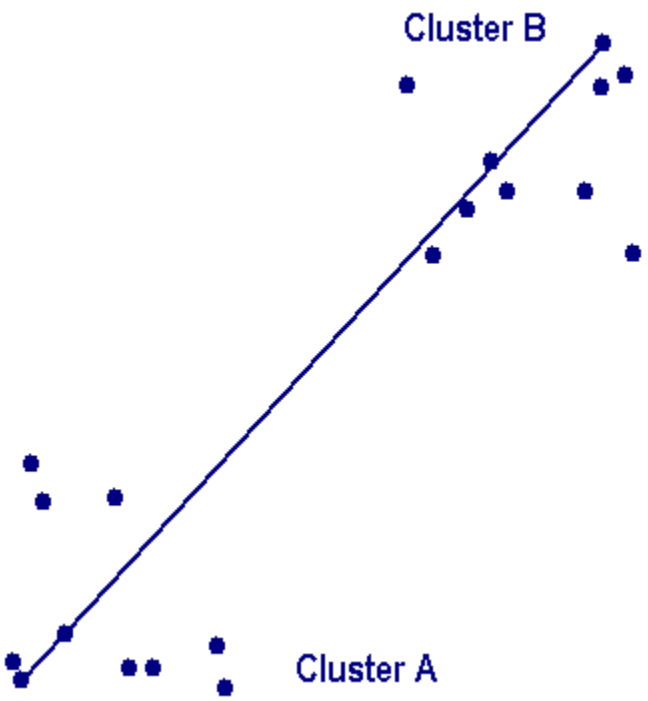
The **Complete Linkage distance** between two clusters is defined as the distance between the most distant (farthest) pair of objects in the two clusters (one object in each cluster). If cluster A is the set of objects $A_1, A_2, \cdots, A_m$ and cluster B is $B_1, B_2, \cdots, B_n$, the Single Linkage distance between clusters A and B is

$D(A,B) = Max\{d_{ij}$: where object A_i is in cluster A and object B_j is in cluster B and d_{ij} is the Euclidean distance between A_i and $B_j\}$

At each stage of hierarchical clustering based on the Complete Linkage distance measure, the clusters A and B, for which D(A, B) is minimum, are merged. The Complete Linkage distance is represented in the XLMINER Help File figure below:

Figure 3

Complete Linkage Distance
Between Clusters



Average Linkage

Under **Average Linkage** the distance between two clusters is defined to be the average of the distances between all pairs of objects, where each pair is made up on one object from each cluster. If cluster A is the set of objects $A_1, A_2, \cdots, A_m$ and cluster B is $B_1, B_2, \cdots, B_n$, the Average Linkage distance between clusters A and B is

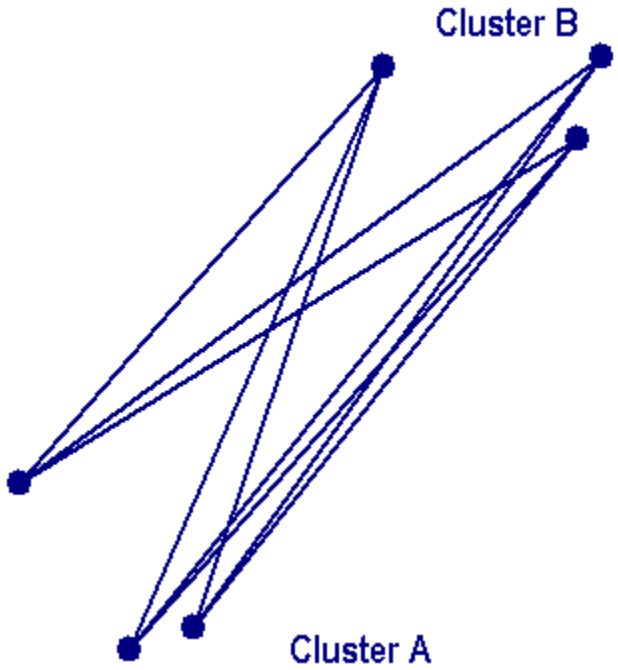
$$D(A,B) = \frac{T_{AB}}{N_A \cdot N_B}$$

where T_{AB} is the sum of all pairwise distances between cluster A and Cluster B. N_A and N_B are the sizes of the clusters A and B, respectively.

At each stage of hierarchical clustering based on the Average Linkage distance measure, the clusters A and B are merged such that, after merger, the average pairwise distance **within the newly formed cluster**, is minimum. The Complete Linkage distance is represented in the XLMINER Help File figure below:

Figure 4

Average Linkage Distance
Between Clusters



Steps in Agglomerative Clustering

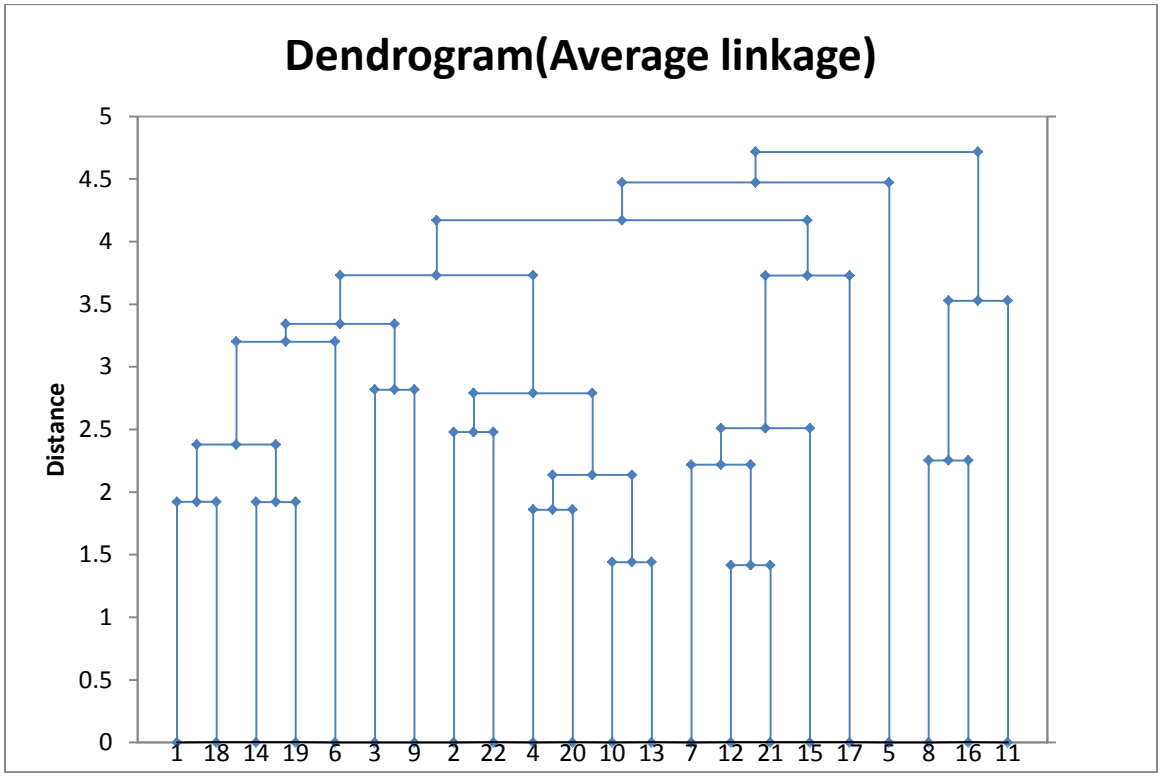
The steps in Agglomerative Clustering are as follows:

1. Start with n clusters (each observation = cluster)
2. The two closest observations are merged into one cluster
3. At every step, the two clusters that are "closest" to each other are merged. That is, either single observations are added to existing clusters or two exiting clusters are merged.
4. This process continues until all observations are merged.

This process of agglomeration leads to the construction of a **dendrogram**. This is a tree-like diagram that summarizes the process of clustering. For any given number of clusters we can determine the records in the clusters **by sliding a horizontal line (ruler) up and down the dendrogram until the number of vertical intersections of the horizontal line equals the number of clusters desired.**

Dendrograms are more useful visually when there are a smaller number of cases as in the Utilities.xls data set. However, the agglomerative procedure works for larger data sets but is computing intensive in that nxn matrices are the basic building blocks for the Agglomerative procedure.

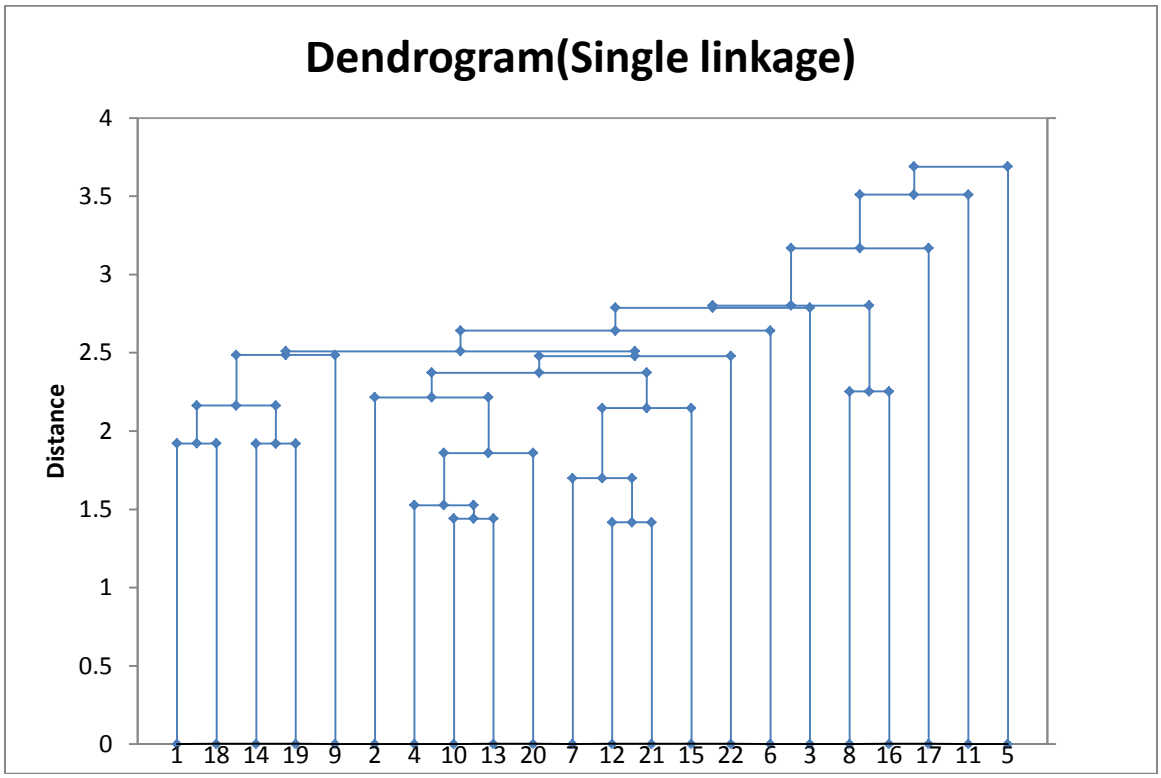
To demonstrate the construction and interpretation of a dendrogram let's cluster the data contained in the Utilities.xls data set. This data set consists of observations on 22 utilities each utility being described by 8 variables. As noted above we have 3 different choices of distance between clusters. They are Single Linkage (Nearest Neighbor), Complete Linkage (Farthest Neighbor) and Average Linkage. Three separate dendrograms can be generated for each choice of distance measure. Let's look at the dendrogram generated by using the Average Linkage measure. It is reproduced below:



If we put our horizontal ruler at 4.0 for the maximal distance allowed between clusters (as measured by average linkage) we "cut across" 4 vertical lines and thus get 4 clusters. They are as follows:{1,18,14,19,6,3,9,2,22,4,20,10,13}; {7,12,21,15,17}; {5} ; {8,16,11}. If we put our horizontal ruler at 3.5 for the maximal distance allowed between clusters we "cut across" 7 vertical lines and thus get 7 clusters. They are as follows: {1,18,14,19,6,3,9}; {2,22,4,20,10,13} ; {7,12,21,15}; {17};{5};{8,16};{11}. The four cluster group is constructed by combining the first and second clusters, the third and fourth clusters, and the sixth and seventh clusters in the seven cluster group. You can now see why this type of clustering is call hierarchical because the 4 cluster group is constructed by combining cluster groupings immediately below it. As you move up

slowly from the bottom of the dendrogram to the top you move from n clusters to n-l clusters to n-2 clusters etc. until all of the observations are contained into one cluster.

To show how sensitive the choice of clusters is to the choice of distance, consider the Single Linkage dendrogram for the Utilities data:



In the case of forming 4 groups, set the maximal allowed distance to be 3.0 in the above dendrogram. Then we get the following 4 clusters:{5} ; {11}; {17}; $\{rest\}$. These four clusters are quite different from the 4 clusters determined by using the Average Linkage dendrogram. **This just goes to show that cluster analysis is an art form and the clusters should be interpreted with caution and hopefully only accepted if the clusters make sense given the domain-specific knowledge we have concerning the utilities under study.**

Also we should note some additional limitations of hierarchical clustering:

- For very large data sets, can be expensive and slow
- Makes only one pass through the data. Therefore, early clustering decisions affect the rest of the clustering results.
- Often has low stability. That is, adding or subtracting variables or adding or dropping observations can affect the groupings substantially.
- Sensitive to outliers and their treatment

Non-hierarchical Clustering (K-means)

The following is hopefully a not too technical discussion of K-means clustering. It is a non-hierarchical method in the sense that if one has 2 clusters, say, generated by pre-specifying 2 means (centroids) in the K-means algorithm and 3 clusters generated by pre-specifying 3 means in the K-means algorithm, then it may be the case that no combination of any two clusters of the 3 cluster group can give rise to the 2 cluster grouping. In this sense the K-means algorithm is non-hierarchical. Let us turn again to the Utilities data and use the K-means clustering method to determine 4 clusters based on the normalized data. We use the following choices:

- Normalized data
- 10 Random Starts
- 10 iterations per start
- Fixed random seed = 12345
- Number of reported clusters = 4

Then the K-means algorithm in XLMiner for four clusters generated the following clusters: cluster 1 = {2,5,7,12,15,17,21,22}; cluster 2 = {4,10,13,20}; cluster 3 = {3,6,9}; cluster 4 = {1,8,11,14,16,18,19}. Again we derive another distinct 4 cluster grouping. Once can then use domain-specific knowledge to determine if this 4 cluster grouping makes more or less sense than the 4 group clusters determined by either of the choices of cluster distance in the agglomerative approach.

The Steps in the K-means Clustering Approach

Given a set of observations $(\mathbf{x_1}, \mathbf{x_2}, \cdots . \mathbf{x_n})$ where each observation is a d-dimensional real vector, then K-means clustering aims to partition the n observations into K sets (K < n), $\boldsymbol{S} = \{S_1, S_2, \cdots . S_K\}$ so as to minimize the within-cluster sum of squares (WCSS):

$$arg\ \min_{\boldsymbol{S}} \sum_{i=1}^{K} \sum_{x_j \in S_i} \left\| \boldsymbol{x_j} - \boldsymbol{\mu_i} \right\|^2 \qquad (1)$$

where $\boldsymbol{\mu_i}$ is the mean of the points in S_i . Now minimizing (1) can, in theory, be done by the **integer programming method** but this can be extremely time-consuming. Instead the **Lloyd algorithm** is more often used. The steps of the Lloyd algorithm are as follows. Given the initial set of K-means $\boldsymbol{m_1^{(1)}}, \cdots, \boldsymbol{m_K^{(1)}}$ which can be specified randomly or by some heuristic, the algorithm proceeds by alternating between two steps:

Assignment Step: Assign each observation to the cluster with the closest mean

$$S_i^{(t)} = \left\{ \boldsymbol{x_j} : \left\| \boldsymbol{x_j} - \boldsymbol{m_i^{(t)}} \right\| \leq \left\| \boldsymbol{x_j} - \boldsymbol{m_{i^*}^{(t)}} \right\| \right\} \text{ for all } i^* = 1,2, \cdots, K. \qquad (2)$$

Update Step: Calculate the **new means** to be the centroids of the observations in the clusters, i.e.

$$\boldsymbol{m_i^{(t+1)}} = \frac{1}{\left|S_i^{(t)}\right|}\sum_{\boldsymbol{x_j}\in S_i^{(t)}} \boldsymbol{x_j} \text{ for } i = 1,2,\cdots,K. \quad (3)$$

Repeat the Assignment and Update steps until WCSS (equation (1)) no longer changes. Then the centroids and members of the K clusters are determined.

Note: When using random assignment of the K-means to start the algorithm, one might try several starting point K-means and then choose the "best" starting point to be the random K-means that produces the smallest WCSS among all of the random starting points K-means tried.

Regardless of the clustering technique used, one should strive to choose clusters that are interpretable and make sense given the domain-specific knowledge that we have about the problem at hand.

- Review Utilities.xls data