Module 9 – August 5, 2015

K-means: Why it's hard to find the absolute minimum SSE.

Recall that we fit k-means by first choosing K initial clusters and then applying the following iterative process until clusters don't move:

- fit data points to existing clusters
- recompute the centroid of each cluster using the mean of fitted data points

Unfortunately, this iterative process gives us local minima of SSE, which depend on our initial selection of K clusters. We use the anaology of dropping a marble into a landscape with multiple hills and valleys: the marble finds the local valley, which may or may not be the lowest valley, globally.

In simple terms, the SSE is the sum of squared distances of each point to its cluster centroid. Mathematically, we characterize SSE with the k-means objective function (which we want to minimize):

k-means Objective function: SSE = $\sum_{j=1}^{k} \sum_{x \in c_j} d(x,\ m_j)^2$

So why is it hard to verify if your minimum SSE is a global minimum?
Because we will never be able to try out all the total number of potential cluster assignments. As the number of data points increases, the number of potential cluster assignments quickly grows, and finding the absolute minimum becomes effectively impossible.

To illustrate this, the number of possible cluster assignments is given by A(n,k):

$$A(n,k) = \frac{1}{k!} \sum_{j=1}^{k} -1^{k-j} \binom{k}{j} j^n$$

$$A(10,4) = 34{,}105\ potential\ cluster\ assignments$$

$$A(25,4) = 5\ * 10^{13}\ potential\ cluster\ assignments$$

Because of the large number of potential clusters and potential for many local minimums within a given observation set, finding the absolute minimum for the k-means objective function is difficult and often unrealistic.

Geometry of k-means

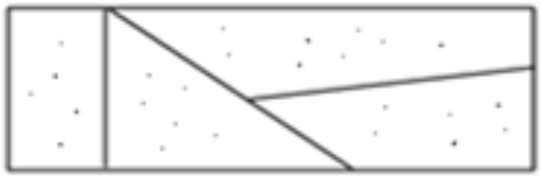
We can think of K-means as not only partitioning data points into clusters, but also as partitioning data points into subsets of Euclidean space. These subsets of Euclidian space are represented by V_j, which is the set of all possible points that are closer to one cluster centroid than another (defined mathematically below). In this section we are trying to show that the subsets of euclidian space determined by k-means will ALWAYS be convex polyhedra.

Mathematical definition for Vj:
With cluster centers, $m_1,\ldots..,m_k$, determined by k-means

$$Let\ V_j = \{x \in R^D : d(x,,c_j) \leq d(x,,c_k)\ for\ all\ k \neq j\}$$

Lets break that definition down. V_j is a set where x can be any possible point in the space, R^D, and R^D occupies the space surround a cluster j, where all points in that space are closer to cluster j than any other cluster, k. The subsets of euclidian space determined by V_j (for each cluster) are shown below:

Convex Polyhedrons in Euclidean Space



Here, each line represents points that are equidistant between two cluster centroids. (Figure is not drawn to scale)

We will always get convex polyheadra.

Definition for convex set: A convex set is made up of points where for any two points, a line drawn between the two points lies within the set. In euclidian space, these convex sets form convex polyhedra.

k-Medioids:

k-medioids is an alternative to k-means clustering. In k-means, centroids are generally not actual observed data points (rather, they are the mean of a set of observed data points). For situations where you need more interpretability from cluster centers (e.g. for images, finding an average of a set as the center may not be meaningful--rather an actual observation may be more interpretable). k-medioids provides an alternative by requiring each cluster center to be an observed data point.

Algorithm:
Repeat 1 and 2 until clusters stop moving:

1. Assign each x_i to the nearest cluster
2. Recalculate each cluster C_j, by minimizing:

$$m \in \{x : x \in c_j\} \sum_{x \in c_j} d(x,m)^2$$

While k-medioids provides better interpretability, the method will result in a higher SSE relative to k-means. This is due to the fact that k-means considers all possible points while k-medioids is constrained to the observed data set. If k-means wanted to choose the point chosen by k-medioids, it could.

In the R package called cluster, pam is the function for k-medioids

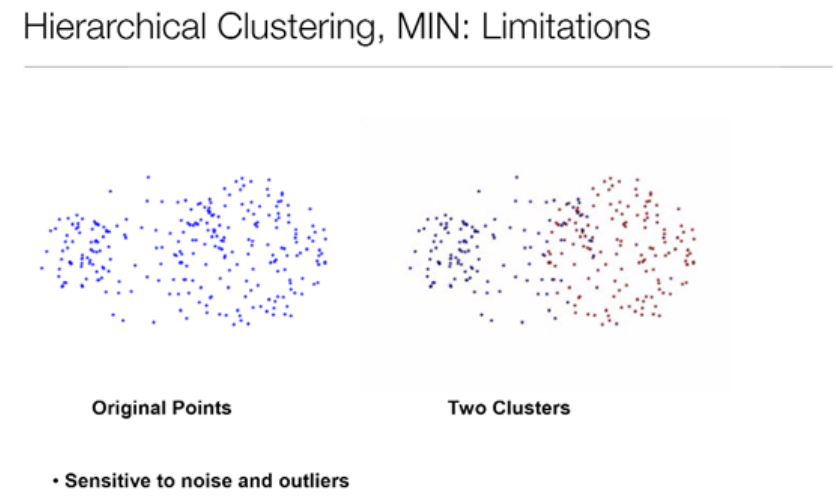
Hierarchical Agglomerative Clustering - Distance Considerations

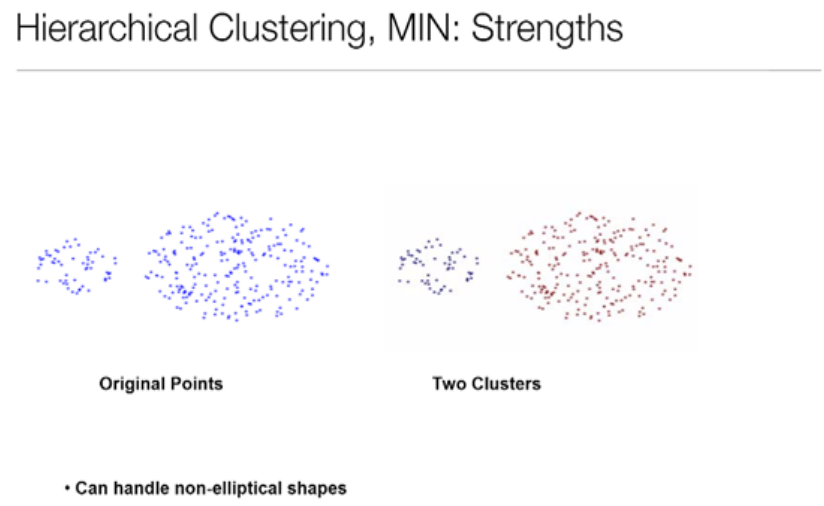
Agglomerative clustering a method of clustering that begins with single data points and progressively creates groupings by merging data points based on the proximity. More information on hierarchical and agglomerative clustering is available in notes from the previous module which discusses these concepts in detail.

In hierarchical agglomerative clustering, a key consideration is how distance between clusters is measured. This influences which clusters are merged through the agglomerative process.

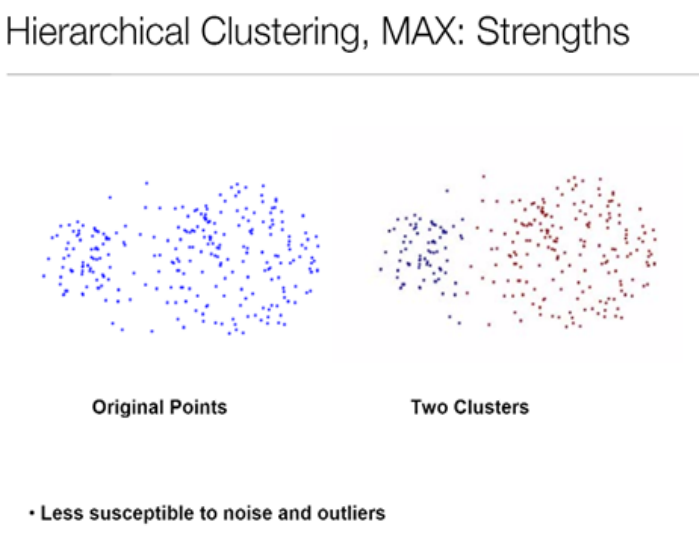
There are three methods of measuring the distances between clusters:

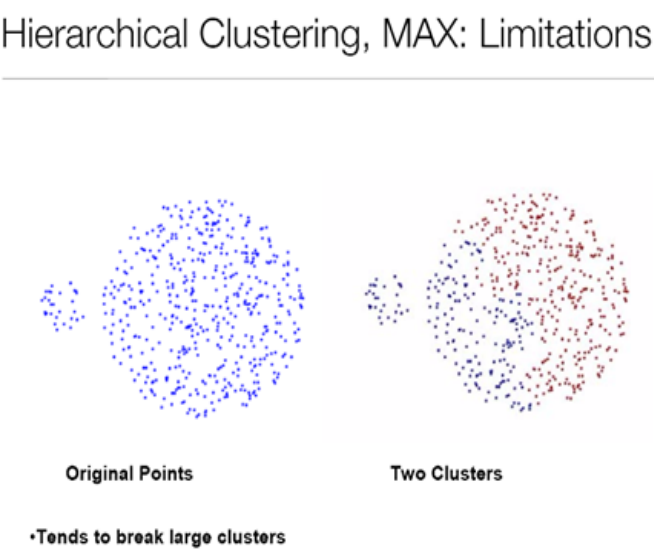
1. Min Linkage ("single" linkage in R) - the distance between two clusters is the distance between the two closest observations between the clusters.
    - Strengths: Min linkage can handle non-elliptical shapes and different sized clusters.
    - Limitations: "Chaining" - only the two closest data points are considered in calculating the distance between clusters and other points in the clusters are not considered. These other points may or may not be most closely related to the cluster they are merged into as a result.





2. Max Linkage ("complete" linkage in R) - the distance between two clusters is the distance between the two furthest observations between the clusters.
    - Strengths: Not as affected by noise and outliers as min linkage.
    - Limitations: "Crowding" - linked points can be closer to points in another cluster than to points in its own cluster. This tends to break up large clusters (see second image below).

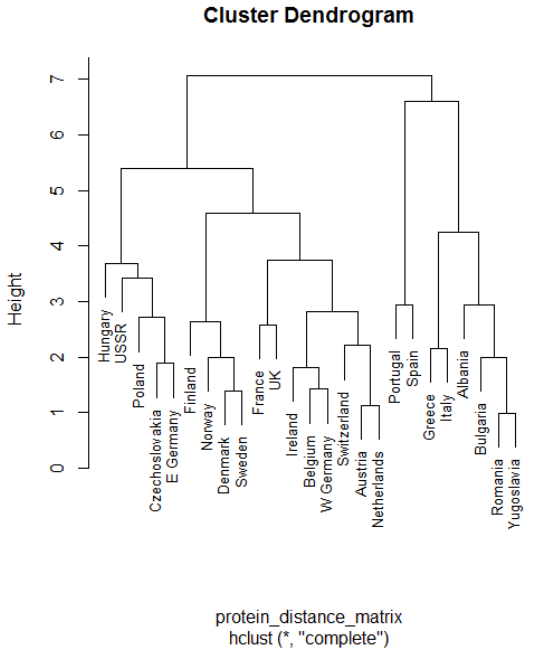


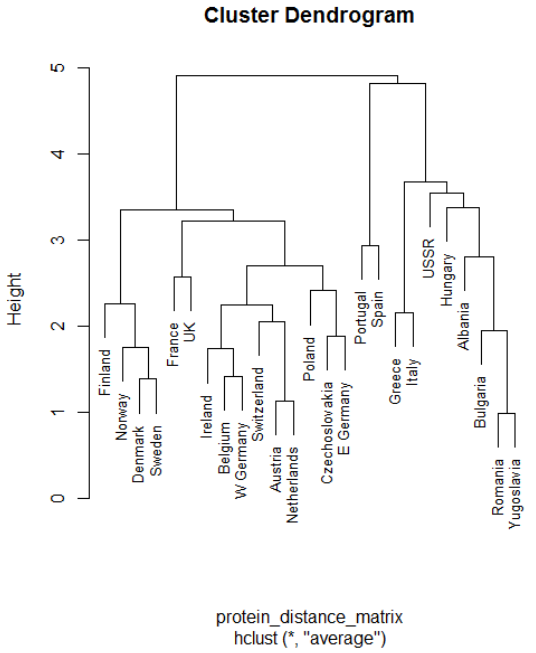


3. Average Linkage - the distance between two clusters is the distance between the average observation in each of the clusters. This attempts to strike a balance between min/max linkage.
    - Limitations: This method is less interpretable. The height between clusters at a given number of k is an average so it can't be said what the min/max difference between two observations within clusters is.

In the dendograms below, height gives the distance, as defined by the method, between two clusters. When using the max method, the height will give the maximum distance between two clusters (and the min method would give the minimum). However, height is less meaningful when using the average method because the min or max distance cannot be determined.

```
par(mfrow=c(1,2))
hier_protein1 = hclust(protein_distance_matrix, method='complete')
hier_protein2 = hclust(protein_distance_matrix, method='average')
plot(hier_protein1, cex=0.8)
plot(hier_protein2, cex=0.8)
```





Selecting k for k-means

In the real world, there may be inputs to guide selection of k, depending on the application of the dataset (for example, you may have certain defined marketing segments, types of consumers, etc.).

One statistical method for selecting k is the CH index. The CH index is defined as follows:

With:

$$Between\ Cluster\ SSE\ -\ B(k) = \sum_{j=1}^{k} n_j\, d\left(n_j,\, \overline{x}\right)^2$$

$$Within\ Cluster\ SSE\ -\ W(k) = \sum_{j=1}^{k} \sum_{x=c_j} d\left(x,\, m_j\right)^2$$

$$CH(k) = \frac{\left(\frac{B(k)}{k-1}\right)}{\left(\frac{W(k)}{n-k}\right)} = \left(\frac{B(k)}{W(k)}\right) * \left(\frac{n-k}{k-1}\right)$$

$$Pick\ \hat{k}\ as\ maximum\ of\ CH(k):$$

$$\hat{k} = argmax(\ k\ \in (2, \dots, k_{max})\ CH(k))$$

$$m_j = cluster\ center\ j$$
$$\overline{x} = \text{grand mean} - sample\ mean\ of\ all\ n\ datapoints$$
$$n_j = \#\ of\ points\ in\ cluster$$

The CH index acts as an "Occams Razor" method of selecting k where the preference is for the most simple solution with all else equal. Simply maximizing B(k) and minimizing W(k) leads to overfit (k would equal n)--so a penalty is introduced for a larger k (the $\left(\frac{n-k}{k-1}\right)$ term).

The CH index is not necessarily optimal and there are alternative statistical methods, such as AIC, BIC, or the gap statistic.