# Hierarchical Clustering

Stephen Vardeman
Analytics Iowa LLC
ISU Statistics and IMSE

# Cluster dissimilarities for hierarchical methods

To apply a hierarchical clustering method, one must first choose a method of using dissimilarities for items to define dissimilarities for clusters. Three common (and somewhat obvious) possibilities in this regard are as follows. For $C_1$ and $C_2$ different elements of a partition of the set of items, or equivalently their $r$ indices, one might define dissimilarity of $C_1$ and $C_2$ as

1. $D(C_1, C_2) = \min\{d_{ij} | i \in C_1 \text{ and } j \in C_2\}$ (this is the "single linkage" or "nearest neighbor" choice),
2. $D(C_1, C_2) = \max\{d_{ij} | i \in C_1 \text{ and } j \in C_2\}$ (this is the "complete linkage" choice), or
3. $D(C_1, C_2) = \frac{1}{\#C_1 \cdot \#C_2} \sum_{i \in C_1,\, j \in C_2} d_{ij}$ (this is the "average linkage" choice).

# An agglomerative hierarchical method

An agglomerative/bottom-up hierarchical clustering algorithm then begins with every item $\mathbf{x}_i$, $i = 1, 2, \ldots, r$ functioning as a singleton cluster. Then one finds the minimum $d_{ij}$ for $i \neq j$ and puts the corresponding two items into a single cluster (of size 2). Then at a stage where there are $m$ clusters, one merges two clusters with minimum dissimilarity make a single cluster, leaving $m - 1$ clusters overall. This continues until there is only a single cluster. The sequence of $r$ different clusterings (with $r$ through 1 clusters) serves as a menu of potentially interesting solutions to the clustering problem. These can be displayed in the form of a dendogram, where cutting the dendogram at a given level picks out one of the (increasingly coarse as the level rises) clusterings. Those items clustered together "deep" in the tree/dendogram are presumably interpreted to be potentially "more alike" than ones clustered together only at a high level.

# A divisive hierarchical method

A divisive/top-down hierarchical algorithm starts with a single "cluster" consisting of all items. One finds the maximum $d_{ij}$ and uses the two corresponding items as seeds for two clusters. One then assigns each $\mathbf{x}_l$ for $l \neq i$ and $l \neq j$ to the cluster represented by $\mathbf{x}_i$ if

$$d(\mathbf{x}_i, \mathbf{x}_l) < d(\mathbf{x}_j, \mathbf{x}_l)$$

and to the cluster represented by $\mathbf{x}_j$ otherwise. When there are $m$ clusters, one identifies the cluster with largest $d_{ij}$ corresponding to a pair of elements in the cluster, splitting it using the method applied to split the original "single large cluster" (to produce an $(m+1)$-cluster clustering). This, like the agglomerative algorithm, produces a sequence of $r$ different clusterings (with 1 through $r$ clusters) that serves as a menu of potentially interesting solutions to the clustering problem. And like the sequence produced by the agglomerative algorithm, this sequence can be represented using a dendogram.

# Thresholding

Both the agglomerative and divisive algorithms may be modified by fixing a threshold $t > 0$ for use in deciding whether or not to merge two clusters or to split a cluster. The agglomerative version terminates when all pairs of existing clusters have dissimilarities more than $t$. The divisive version terminates when all dissimilarities for pairs of items in all clusters are below $t$. Employing a threshold has the potential to shorten the menu of clusterings produced by either of the methods to include less than $r$ clusterings. (Thresholding the agglomerative method cuts off the top of the corresponding full dendogram, and thresholding the divisive method cuts off the bottom of the corresponding full dendogram.)