Hierarchical Clustering

Eric Ropeta - MTH496

29 April 2022

When labeled data is sparse or not present at all, it is often useful to place the data into groups with other nearby or similar elements. The process of grouping similar points into dissimilar groups is called clustering, and there are many different approaches to clustering that are often tailored to different structures of data. One of the most popular clustering algorithms is *k*-means clustering, which is a centroid-based clustering approach that aims to represent each cluster by its mean point, or centroid. The algorithm then assigns new data to the clusters with the nearest centroid. *k*-means clustering, however, finds difficulty performing when the clusters have different or unclear shapes, sizes, or dimensions. Another disadvantage is the requirement to pre-specify an optimal *k* value. One method of clustering that circumvents these issues is called hierarchical clustering, which is named in reference to the nesting of clusters within other clusters. Hierarchical clustering, according to Murtagh (1985), saw much of its early work in the field of biological taxonomy in the 1950s and 1960s. It was a common practice to model evolutionary relationships between organisms in a hierarchical fashion, and it is hoped to remain an accurate model for the progression of evolution. Another useful implementation of hierarchical clustering is in document analysis, where one might aim to sort documents based on the words and phrases contained within a document. Similarly, words and phrases can be clustered by common usage with other words in an effort to sort text by language or any other means.

Generally, hierarchical data structures can be visually represented by a tree-like graph called a dendrogram. The root node, located at the top of the dendrogram, represents the data set as a whole, and the leaves depict the individual data points at the bottom of the figure. Meanwhile, the intermediary nodes describe the extent to which objects are similar to each other, and the dendrogram's height corresponds to the distance between the points (Xu, 2008). Hierarchical clustering approaches are either agglomerative or divisive, with agglomerative being the most widely used. Agglomerative clustering is a "bottom-up" approach in regards to the dendrogram, meaning that each data point begins as its own cluster and similar clusters fuse together as the algorithm progresses. Agglomerative clustering offers great generality, as there are many linkage criteria and distance metrics that can tailor the model to the structure of the data. Furthermore, divisive clustering uses a 'top-down' approach, so the entire dataset starts as one all-inclusive cluster. Clusters are then successively split into dissimilar clusters until there are as many clusters as there are data points. Additionally, there are several algorithms that are based on hierarchical clustering that will be discussed further on. Overall, hierarchical clustering provides a useful solution to clustering complex data structures.

The general algorithm for agglomerative clustering is known to some as the Sequential Agglomerative Hierarchical Non-overlapping (SAHN) algorithm. The algorithm begins at the bottom of the dendrogram, where there exist *n* clusters and each cluster contains exactly one point. Step 1 is to a) choose an initial cluster C_1 and b) initialize a proximity matrix *P*. The proximity matrix is defined by either a similarity matrix or a distance matrix that contains the proximity between C_1 and all other clusters. Step 2 is to find the best cluster to merge with C_1. One can accomplish this by finding the minimum value of P, which corresponds to the closest or most similar cluster that one seeks. Step 3 is to merge the two clusters, and Step 4 is to update *P* to reflect the proximities between the newly formed cluster and the remaining clusters (Embrechts et al., 2013). Repeat Steps 2, 3, and 4 until the entire data set is forced into one cluster. Finally, cut the reslulting dendrogram with a horizontal line at a height

corresponding to the desired amount of clusters to observe, as shown in Figure 1 (Xu and Wunsch, 2008).

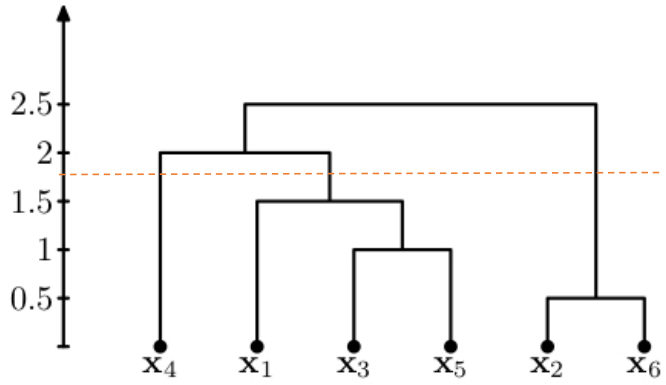

Figure 1. *a dendrogram cut at* $y = 1.75$ *to display 3 clusters (Gan et al., 2007)*

As seen in the SAHN algorithm, agglomerative clustering heavily depends on the definition of proximity between clusters. Furthermore, this definition of distance or similarity can be generalized by using Lance and Williams' proposed recursive formula (1967) and notation by Embrechts et al. (2013): The distance between a cluster C_l and a newly formed cluster C_{ij} from the merge of C_i and C_j is defined as

$$\mathrm{D}(C_l, (C_i, C_j)) = \alpha_\mathrm{i}\mathrm{D}(C_l, C_i) + \alpha_\mathrm{j}\mathrm{D}(C_l, C_j) + \beta\mathrm{D}(C_i, C_j) + \gamma|\mathrm{D}(C_l, C_i) - \mathrm{D}(C_l, C_j)|$$

where $\mathrm{D}(*, *)$ is any distance function, such as the Euclidean distance, Manhattan distance, and the cosine similarity metric. The set of parameters $\{\alpha_\mathrm{i}, \alpha_\mathrm{j}, \beta, \gamma\}$ defines the linkage method used in defining distance. That is, the linkage criteria determines from which point on C_{ij} to measure the distance from. For example, the single-linkage method has criteria $\{1/2, 1/2, 0, -1/2\}$. This method defines proximity to be the *minimal distance* between any pair of points where each part of the pair comes from a distinct cluster. Using the parameter definitions in the table above, Lance and Williams' recursive formula becomes

$$\mathrm{D}(C_l, (C_i, C_j)) = \min(\mathrm{D}(C_l, C_i), \mathrm{D}(C_l, C_j))$$

Single linkage works well for clusters that are far apart or highly dissimilar (Xu and Wunsch, 2008). This method often generates elongated clusters, and this can lead to the chaining effect and sensitivity to noise. According to Murtagh (1985), chaining occurs when two seemingly unrelated objects become clustered together due to an intermediary similarity. An example of this can occur when the general disciplines "Art" and "Engineering" end up in a cluster together due to the subdiscipline of both, "Architecture". A similar linkage method with quite different results than single-linkage is the complete-linkage method with linkage criteria $\{1/2, 1/2, 0, 1/2\}$. This method utilizes the *maximal* distance between any pair of points from different clusters, and it becomes useful when one needs to identify small, compact clusters. Lance and Williams' formula for the complete linkage criteria is

$$\mathrm{D}(C_l, (C_i, C_j)) = \max(\mathrm{D}(C_l, C_i), \mathrm{D}(C_l, C_j))$$

Furthermore, the group average linkage method uses, as its definition of distance, the average distance between every pair of points that come from two distinct clusters in question. Similarly, the weighted average linkage method uses the same definition as the group average linkage method with the exception that the distance between the newest cluster and the rest of the clusters is weighted by the number of points in each cluster.

Three more linkage methods worth note are the centroid linkage method, the median linkage method, and Ward's method. The centroid linkage method considers the centroid of the cluster and defines distance as the distance between the two centroids of the clusters in question. The centroid is calculated by $\boldsymbol{m}_{\mathrm{i}} = \frac{1}{n_i}\sum_{x \epsilon C_i} x$ and Lance and Williams' (1967) recursive formula becomes

$$\mathrm{D}(C_l, (C_i, C_j)) = \frac{\mathrm{n_i}}{\mathrm{n_i} + \mathrm{n_j}}\mathrm{D}(C_l, C_i) + \frac{\mathrm{n_j}}{\mathrm{n_i} + \mathrm{n_j}}\mathrm{D}(C_l, C_j) - \frac{\mathrm{n_i n_j}}{(\mathrm{n_i} + \mathrm{n_j})^2}\mathrm{D}(C_i, C_j)$$

According to Hastie et al. (2013), the formula above holds equivalence with the squared Euclidean distance between the centroids of two clusters: $\mathrm{D}(C_l, (C_i, C_j)) = \|\mathbf{m}_{\mathrm{l}} + \mathbf{m}_{\mathrm{ij}}\|^2$. Furthermore, the median linkage method is simply a special case of the centroid linkage method where the merging clusters have equal size, so equal weight is given to them. Ward's method seeks to minimize the loss of information that occurs with each merging of clusters. This loss of information is quantified by the sum of squares error

$\mathrm{E} = \sum_{\mathrm{k}=1}^{\mathrm{K}} \sum_{\mathrm{x_j} \epsilon \mathrm{C_k}} \|\mathbf{x}_{\mathrm{i}} - \mathbf{m}_{\mathrm{k}}\|^2$, where there exist *K* clusters and $\boldsymbol{m}_k$ is the centroid of cluster C_k. The change in the sum of squares error between clusters is denoted by $\Delta \mathrm{E}_{\mathrm{ij}} = \frac{\mathrm{n_i n_j}}{\mathrm{n_i}+\mathrm{n_j}}\|\mathbf{m}_{\mathrm{i}} - \mathbf{m}_{\mathrm{j}}\|^2$. Hence, a given cluster will merge with the cluster that gives the minimal increase in this error (Gan et al., 2007). The recursive formula is then written as

$$\mathrm{D}(C_l, (C_i, C_j)) = \frac{\mathrm{n_i} + \mathrm{n_l}}{\mathrm{n_i} + \mathrm{n_j} + \mathrm{n_l}}\mathrm{D}(C_l, C_i) + \frac{\mathrm{n_j} + \mathrm{n_l}}{\mathrm{n_i} + \mathrm{n_j} + \mathrm{n_l}}\mathrm{D}(C_l, C_j) - \frac{\mathrm{n_l}}{(\mathrm{n_i} + \mathrm{n_j})^2}\mathrm{D}(C_i, C_j)$$

According to Xu and Wunsch (2008), single linkage, complete linkage, and each average linkage method are considered graph methods due to their consideration of every point in each pair of clusters. Meanwhile, the centroid method, median method, and Ward's method are considered geometric methods since they use geometric centers instead of actual points in a given cluster.

In contrast to agglomerative hierarchical clustering, the divisive clustering algorithm is known as DIANA (DIvisive ANAlysis). It begins at the top of the dendrogram where the entire data set is encompassed into one cluster. Gan et al. (2007) describe two types of divisive clustering that affect the process of splitting the clusters. Monothetic divisive clustering indicates that the division of data is determined by a cluster possessing one specified attribute. Meanwhile, polythetic divisive clustering determines how the data is split by considering values under all attributes. Both types of divisive clustering divide the cluster with the largest diameter in each iteration of the algorithm. Note that the diameter of a cluster is the maximum proximity between two points within the cluster. This splitting occurs in succession until each cluster contains exactly one point. To define the algorithm formally, consider a cluster C of size $|\mathrm{C}| \geq 2$, and let clusters A and B be the clusters derived from C. Then, initialize $\mathrm{A} = \mathrm{C}$ and $\mathrm{B} = \Phi$, where Φ is the whole data set. A and B are computed by iteratively placeing points from A into B by performing the following operation: the first point will be moved from A to B if it maximizes the function $\mathrm{D}(\mathbf{x}, \mathrm{A} \setminus \{\mathbf{x}\}) = \frac{1}{|\mathrm{A}|-1}\sum_{\mathbf{y} \in \mathrm{A}, \mathrm{y} \neq \mathrm{x}} \mathrm{d}(\mathbf{x}, \mathbf{y})$ where $\mathrm{d}(*, *)$ is any distance function that fits the context of the situation (Gan et al., 2007). If the first point does indeed maximize that function, then A and B can be updated to $\mathrm{A}_{\mathrm{new}} = \mathrm{A}_{\mathrm{old}} \setminus \{\mathbf{y}\}$ and $\mathrm{B}_{\mathrm{new}} = \mathrm{B}_{\mathrm{old}} \cup \{\mathbf{y}\}$. Then, the algorithm looks for other points in A that should move to B by using the test function

$$\mathrm{D}(\mathbf{x}, \mathrm{A} \setminus \{\mathbf{x}\}) - \mathrm{D}(\mathbf{x}, \mathrm{B}) = \frac{1}{|\mathrm{A}| - 1}\sum_{\mathbf{y} \in \mathrm{A}, \mathrm{y} \neq \mathrm{x}} \mathrm{d}(\mathbf{x}, \mathbf{y}) - \frac{1}{|\mathrm{B}|}\sum_{\mathrm{z} \in \mathrm{B}} \mathrm{d}(\mathbf{x}, \mathbf{z})$$

where if a point maximizes that function and if that maximal value is strictly positive, then the point will be moved from A to B. Contrarily, if the maximal value is nonpositive, then the process halts, and then that division is complete (Gan et al., 2007). DIANA is less commonly used when compared to agglomerative clustering, and that is largely due to the heavy cost of computation. In this algorithm, the first step alone involves performing $2^{N-1} - 1$ computations to find the proximities for every possible division of the data into two nonempty subsets (Xu and Wunsch, 2008). However, it can still provide a clear insight into the true structure of the data, and it continues to be used for a variety of applications today.

Embrechts et al. (2013) explains a number of advantages to hierarchical clustering, including its aptitude to recursion and its lack of stochastic elements. Furthermore, hierarchical clustering is able to exploit many application-specific metrics, and it can be well-illustrated using a dendrogram. However, there exist several drawbacks to hierarchical clustering as well. For example, the algorithm increasingly performs worse as the size of the data becomes larger. This is due to the nature of iteratively calculating the proximity matrix - memory and computation time suffer as a result. Moreover, the assumption of hierarchical structure in data is often unrealistic for any given data set. One final disadvantage of hierarchical clustering is that performing the algorithm on one data set multiple times while varying the linkage criteria or the distance metric can produce severely different results, as shown in Figure 2 below.

To continue, an important aspect of all machine learning models is the ability to evaluate a model's performance. Embrechts et al. (2013) discuss two types of ways (indices) that compare clusterings with one another. The first are internal cluster validation indices, and these do not rely on the presence of external class labels. Some examples of internal cluster validation indices are the Silhouette width validity index, the Davies-Bouldin index, and the GAP statistic. The Davies-Bouldin index, however, is that of a modified nature since its original purpose was for *k*-means, where the clusters are based on a pre-defined amount of centroids. It has been known to omit the concept of a centroid from a new definition of the Davies-Bouldin index all together. The Silhouette width validity index is based on the width of each cluster as if one was looking at its silhouette. This index evaluates how tight the clusters are along with the optimal number of clusters contained in the data. Furthermore, the GAP statistic is another method of estimating the proper number of clusters in a data set. This statistic considers the cluster dispersion that results from an algorithm, and it compares the dispersion to that of a randomly generated set of data in the same space with the same number of data points and attributes as the original dispersion (Embrechts et al., 2013). In contrast, the second type of indices that compare clusterings are external cluster validation indices. As an example, the Rand index can be utilized to determine how well a given clustering corresponds to pre-assigned labels. Second is the Jaccard coefficient, which is based on a contingency table for a cluster. It can indicate how well a cluster corresponds to pre-assigned labels as well as comparing clustering quality between different clusterings. The indices that have been introduced are but a small number of ways to analyze performance of clustering models, and many of the existent indices are extensions of those that were just discussed.

In conclusion, hierarchical clustering is a powerful tool that can provide insight into unclear data structures and classification of complex data. Hierarchical clusterings can be represented by a dendrogram that shows nested groupings of clusters. Furthermore,

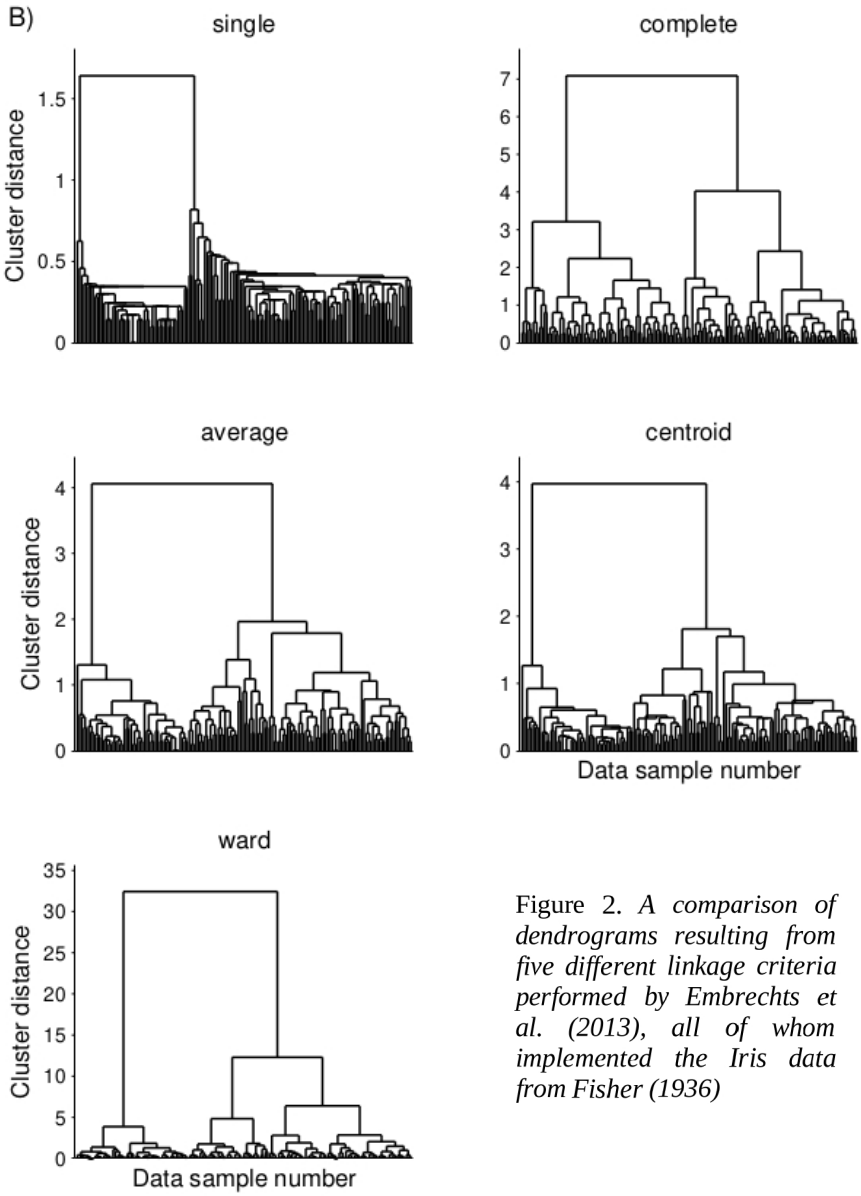

Figure 2. *A comparison of dendrograms resulting from five different linkage criteria performed by Embrechts et al. (2013), all of whom implemented the Iris data from Fisher (1936)*

agglomerative clustering presented itself to be a versatile algorithm with many areas of application from biology to document analysis. Agglomerative clusterings also use a variety of linkage criteria and distance metrics that tailor the algorithm to the type of data under investigation. Moreover, divisive hierarchical clustering provides a more clear insight into the actual structure of the data, but does so at the cost of computation time and memory usage. Finally, there exist methods of evaluating the performance of a model or comparing the performance of multiple models. These include internal cluster validation indices as well as external cluster validation indices. It is hoped that hierarchical clustering can continue to be implemented in solving complex issues, just as its versatility launched it into popularity today.

References

Embrechts, M. J., Gatti, C. J., Linton, J., Roysam, B. (2013). Hierarchical clustering for large data sets. *Springer-Verlag Berlin Heidelberg.* https://doi.org/10.1007/978-3-642-28696-4_8

Fisher, R.A. (1936): The Use of Multiple Measurements in Axonomic Problems. *Ann. Eugen.* 7, pp. 179–188.

Gan, G., Ma, C., Wu, J. (2007). Data clustering: Theory, algorithms, and applications. *ASA-SIAM Series on Statistics and Applied Probability*. https://doi.org/10.1137/1.9780898718348

Hastie, T., James, G., Tibshirani, R., & Witten, D. (2013). An introduction to statistical learning: With applications in R. *Springer New York Heidelberg Dordrecht London.* 10.1007/978-1-4614-7138-7.

Lance, G. N., Williams, W. T. (1967). A General Theory of Classificatory Sorting Strategies: 1. Hierarchical Systems. *The Computer Journal*, 9(4), pp. 373-380. https://doi.org/10.1093/comjnl/9.4.373

Murtagh, F. (1985), Multidimensional Clustering Algorithms, Physica-Verlag, Wurzburg.

Xu, R., & Wunsch II, D. C. (2008). Clustering. *IEEE Press Series on Computational Intelligence*, pp. 31-62.