Hierarchical

#cluster #StatisticalModeling

Hierarchical Cluster Analysis

Introduction[1]

- In a hierarchical classification the data are not partitioned into a particular number of classes or clusters at a single step. Instead the classification consists of a series of partitions, which may run from a single cluster containing all individuals, to *n* clusters each containing a single individual.
- Hierarchical clustering techniques can be subdivided into two methods. Each operates on a proximity matrix[2] of some kind.
  - Hierarchical *agglomerative* (HA) methods, which proceed by a series of successive fusions of the *n* individuals into groups
  - Hierarchical *divisive* (HD) methods, which separate the *n* individuals successively into finer groupings
- Divisions or fusions of clusters, once made, are irrevocable
  - When an HA has joined two individuals, they cannot subsequently be separated. When an HD has made a split it cannot be undone.
- The HA techniques will ultimately reduce the data to a single cluster containing all the individuals. The HD techniques will finally split the entire set of data into n groups each containing a single individual.
  - The investigator wishing to have a solution with an "optimal" number of clusters will need to decide when to stop.
- Hierarchical classifications produces by either technique may be represented by a 2D diagram known as a *dendrogram*, which illustrates the fusions or divisions made at each stage of the

analysis.

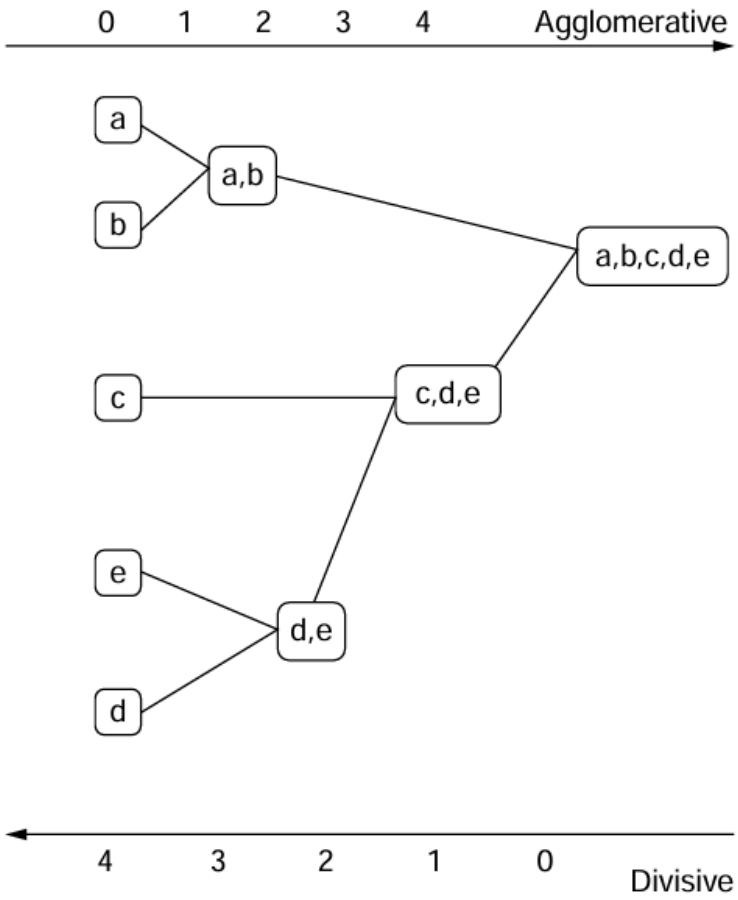


Figure 4.1 *Example of a hierarchical tree structure. (* Taken from *Finding Groups in Data,* 1990, Kaufman and Rousseeuw, with permission of the publisher, John Wiley & Sons, Inc.).

Agglomerative methods

- Agglomerative procedures are probably the most widely used of the hierarchical methods
- They produce a series of partitions of the data:
  - *n* single-member clusters
  - a single group containing all n individuals
- The basic operation of all such methods is similar, and will be illustrated for two specific examples, single linkage and centroid linkage. At each stage the methods fuse individuals or groups of individuals which are closest (or most similar)
- Differences between the methods arise because of the different ways of defining distance (or similarity) between an individual and a group containing several individuals, or between two groups of individuals[2-1]

Illustrative examples of agglomerative methods

- In this section, two hierarchical techniques are illustrated, the first requiring solely a proximity matrix, the second requiring access to a data matrix

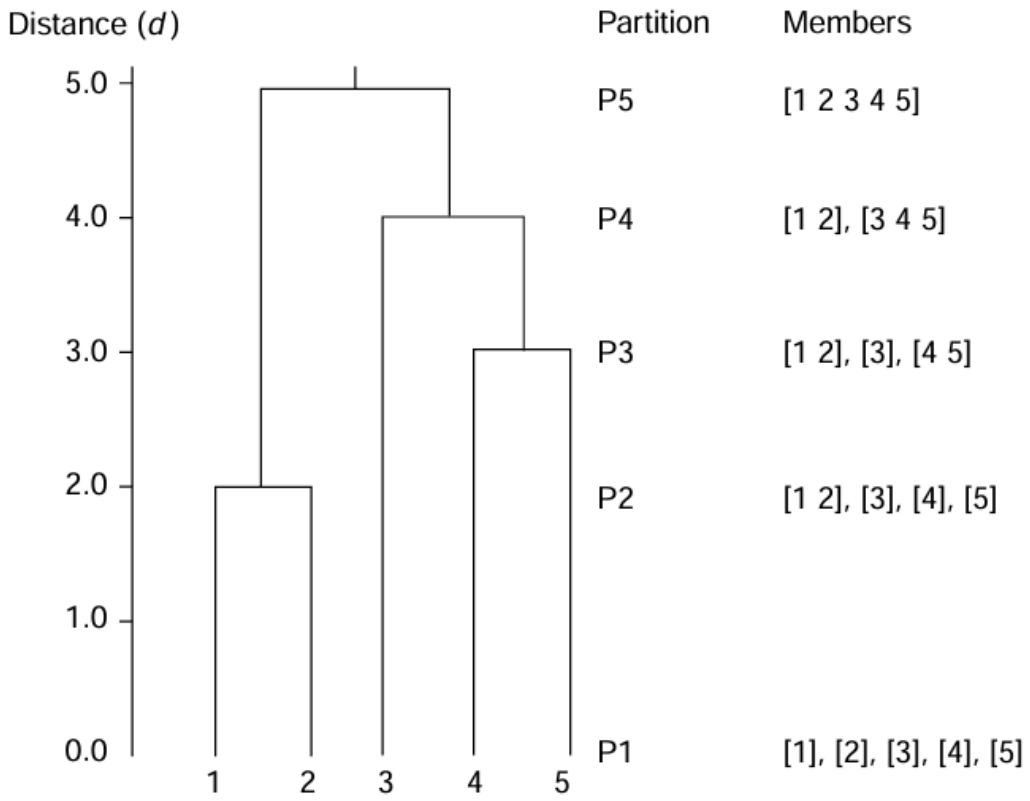


Figure 4.2 *Dendrogram for worked example of single linkage, showing partitions at each step.*

- The first illustration is of one of the simplest hierarchical clustering methods, *single linkage*, also known as the nearest-neighbor technique
  - The defining feature of the method is that the distance between groups is defined as that of the closest pair of individuals, where only pairs consisting of one individual from each group are considered[2-2]
  - Single linkage serves to illustrate the general procedure of a hierarchical method, and in the example below it is applied as an agglomerative method.
    - However, it could equally well be applied as a divisive method, by starting with a cluster containing all objects and then splitting into two clusters whose nearest neighbor distance is a maximum
  - Single linkage operates directly on a proximity matrix
- Another type of clustering, *centroid clustering*, requires access to the original data.
- An important point to note about the two methods mentioned above is that the clustering proceed hierarchically, each being obtained by the merger of clusters from the previous level.
  - So, for example, in neither of the examples above could clusters (1,2,4) and (3,5) have been formed, since neither is obtainable by merging existing clusters.

The standard agglomerative methods

- In addition to those introduced in the previous section, there are several other possible inter-group proximity measures[2-3], each giving rise to a different agglomerative method
- *Complete linkage* (or furthest neighbor) is opposite to single linkage, in the sense that distance between groups is now defined as that of the most distant pair of individuals

- In *group average linkage* - also known as the unweighted pair - group method using the average approach (UPGMA)– the distance between two clusters is the average of the distance between all pairs of individuals that are made up of one individual from each group

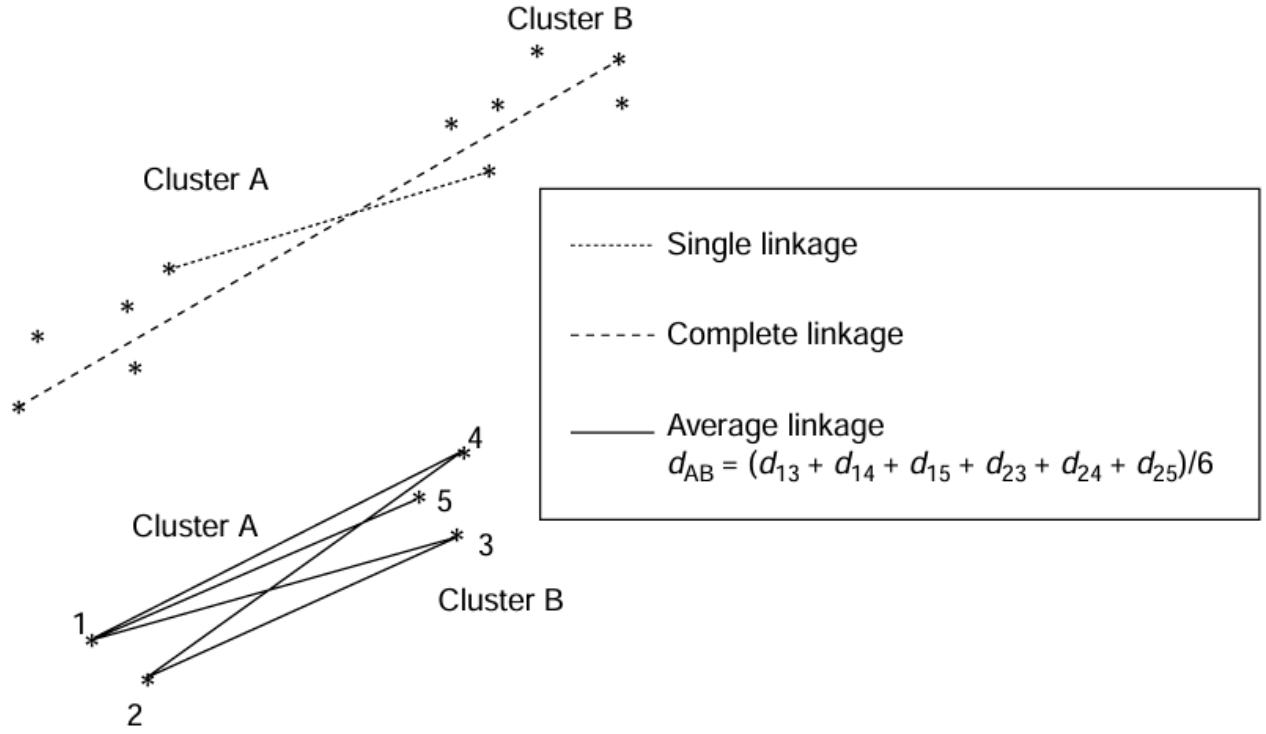


Figure 4.3 *Examples of three inter-cluster distance measures: single, complete and average.*

- All these three methods (single, complete and average) use a proximity matrix as input, and the inter-cluster distances they use are each illustrated graphically in Figure4.3
- Another agglomerative hierarchical method is *centroid clustering* – also known as the unweighted pair - group method using the centroid approach (UPGMC) which uses a data matrix rather than a proximity matrix and involves merging clusters with the most similar mean vectors.
- *Median linkage*– the weighted pair-group method using the centroid approach (WPGMC)– is similar, except that the centroids of the constituent clusters are weighted equally to produce the new centroid of the merged cluster.
    - This is to avoid the objects in the more numerous of the pair of clusters to be merged dominating those in the smaller cluster. The new centroid is thus intermediate between the two constituent clusters.

Standard agglomerative hierarchical clustering methods:

Method	Alternative name	Usually used with	Distance between clusters defined as:	Remarks
Single linkage (Sneath, 1957)	Nearest neighbour	Similarity or distance	Minimum distance between pair of	Tends to produce unbalanced and straggly clusters ('chaining'), especially in large

Method	Alternative name	Usually used with	Distance between clusters defined as:	Remarks
			objects, one in one cluster, one in the other	data sets. Does not take account of cluster structure.
Complete linkage (Sorensen, 1948)	Furthest neighbor	Similarity or distance	Maximum distance between pair of objects, one in one cluster, one in the other	Tends to find compact clusters with equal diameters (maximum distance between objects). Does not take account of cluster structure.
Group/Average linkage (Sokal and Michener, 1958)	UPGMA	Similarity or distance	Average distance between pair of objects, one in one cluster, one in the other	Tends to join clusters with small variances. Intermediate between single and complete linkage. Takes account of cluster structure. Relatively robust.
Centroid linkage (Sokal and Michener, 1958)	UPGMA	Distance (requires raw data)	Squared Euclidean distance between mean vectors (centroids)	Assumes points can be represented in Euclidean space (for geometrical interpretation). The more numerous of the two groups clustered dominates the merged cluster. Subject to reversals.
Weighted average linkage (McQuitty, 1966)	WPGMA	Similarity or distance	Average distance between pair of objects, one in one cluster, one in the other	As for UPGMA, but points in small clusters weighted more highly than points in large clusters (useful if cluster sizes are likely to be uneven).
Medium linkage (Gower, 1967)	WPGMA	Distance (requires raw data)	Squared Euclidean distance between weighted centroids	Assumes points can be represented in Euclidean space for geometrical interpretation. New group is intermediate in position between merged groups. Subject to reversals.
Ward's method (Ward, 1963)	Minimum sum of squares	Distance (requires raw data)	Increase in sum of squares within clusters, after fusion, summed over all variables	Assumes points can be represented in Euclidean space for geometrical interpretation. Tends to f ind same-size, spherical clusters. Sensitive to outliers.

Problems of agglomerative hierarchical methods

- To illustrate some of the potential problems of these agglomerative methods, a set of simulated data will be clustered using single, complete and average linkage. The data consist of 50 points simulated from two bivariate normal distributions with mean vectors (0, 0) and (4, 4), and common covariance matrix

$$\sum = \begin{pmatrix} 16.0 & 1.5 \\ 1.5 & 0.25 \end{pmatrix}$$

- Two intermediate points have been added for the first analysis, in order to illustrate a problem known as *chaining* often found when using single linkage. Figure 4.4 gives the single linkage dendrogram and Figure 4.5 shows some of the results of the cluster analyses.
  - Figure 4.4 shows a typical single linkage dendrogram. There is little clear structure, with the two intermediate points (51 and 52) linking the two main clusters, which are gradually pulled together into one large cluster, isolating two singletons until the final step. Note that although the outlying points 8 and 29 are close together on the dendrogram, they are those at the extreme opposite ends of the main clusters. Note also that this form of dendrogram places the labels for the points just underneath the place where they first join a cluster. This makes the order of joining evident. Some

other software would place all labels along the zero line at the bottom.

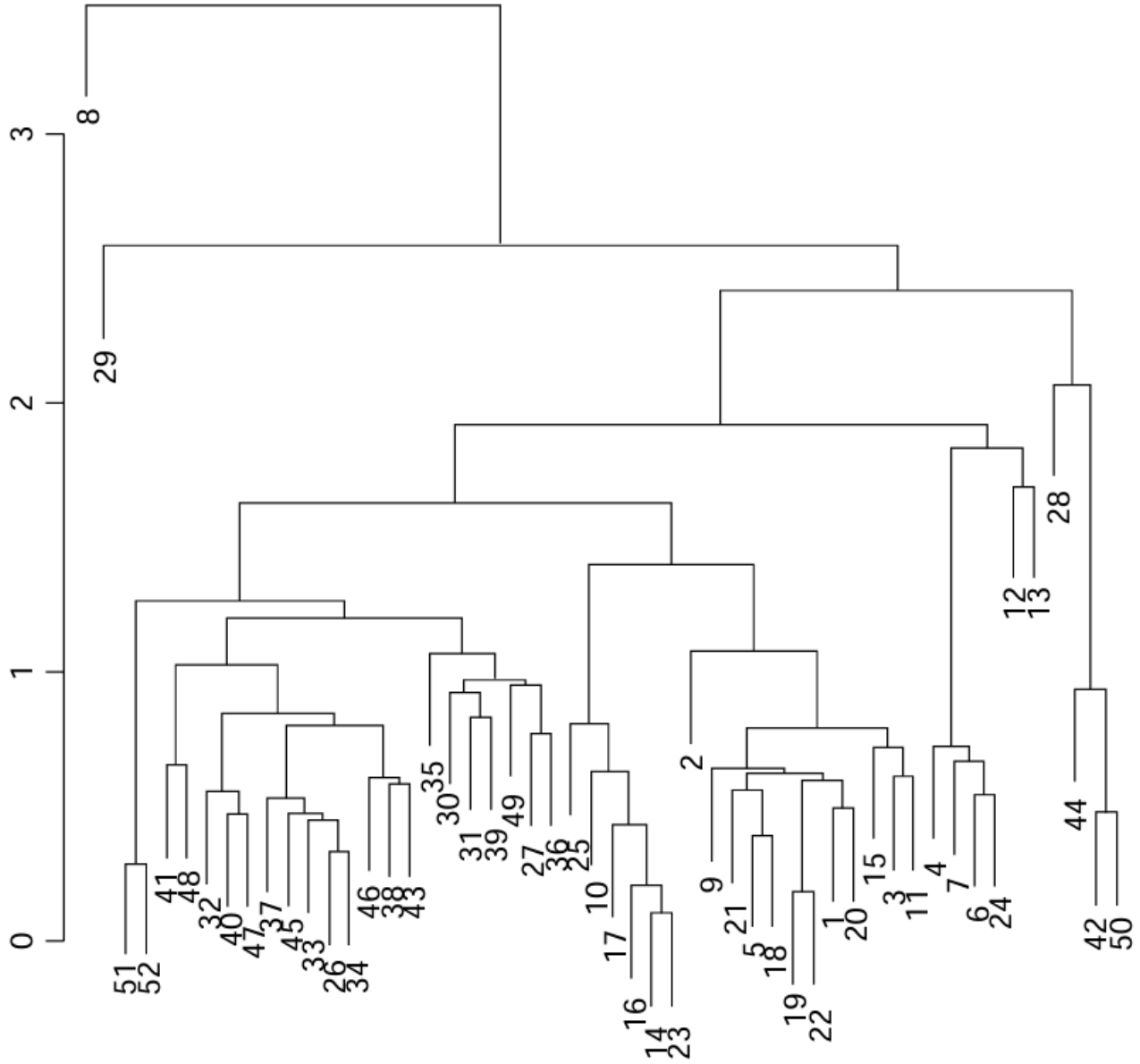


Figure 4.4 *Dendrogram showing single linkage clustering of simulated data set (see also Figure 4.5(a)).*

- Figure 4.5(a) shows the chaining of the two main groups together in single linkage, and the isolation of one outlier, if two groups are specified. (If three groups are specified, the other outlier is hived off, still leaving one large cluster.) Despite the obvious lack of success in recovering the two groups, this example does illustrate a potential benefit of applying single linkage, namely that it can be used to identify outliers, since these are left as singletons if they are sufficiently far from their nearest neighbor.
  - Complete (Figure 4.5(c)) and average linkage (Figure 4.5(d)) techniques were equally unsuccessful in cluster recovery, with or without intermediate points, and whatever number of clusters was specified (from two to five). They tended to impose spherical clusters, forming a cluster in the middle, part from group 1 and part from group 2. The five-cluster solution for single linkage (Figure 4.5(b)) was relatively more successful, since the five clusters could be amalgamated into two, to form the correct groups. (Of course, such an amalgamation would destroy the hierarchy,

just as in the Hawkins et al. example given in Section 4.1.)

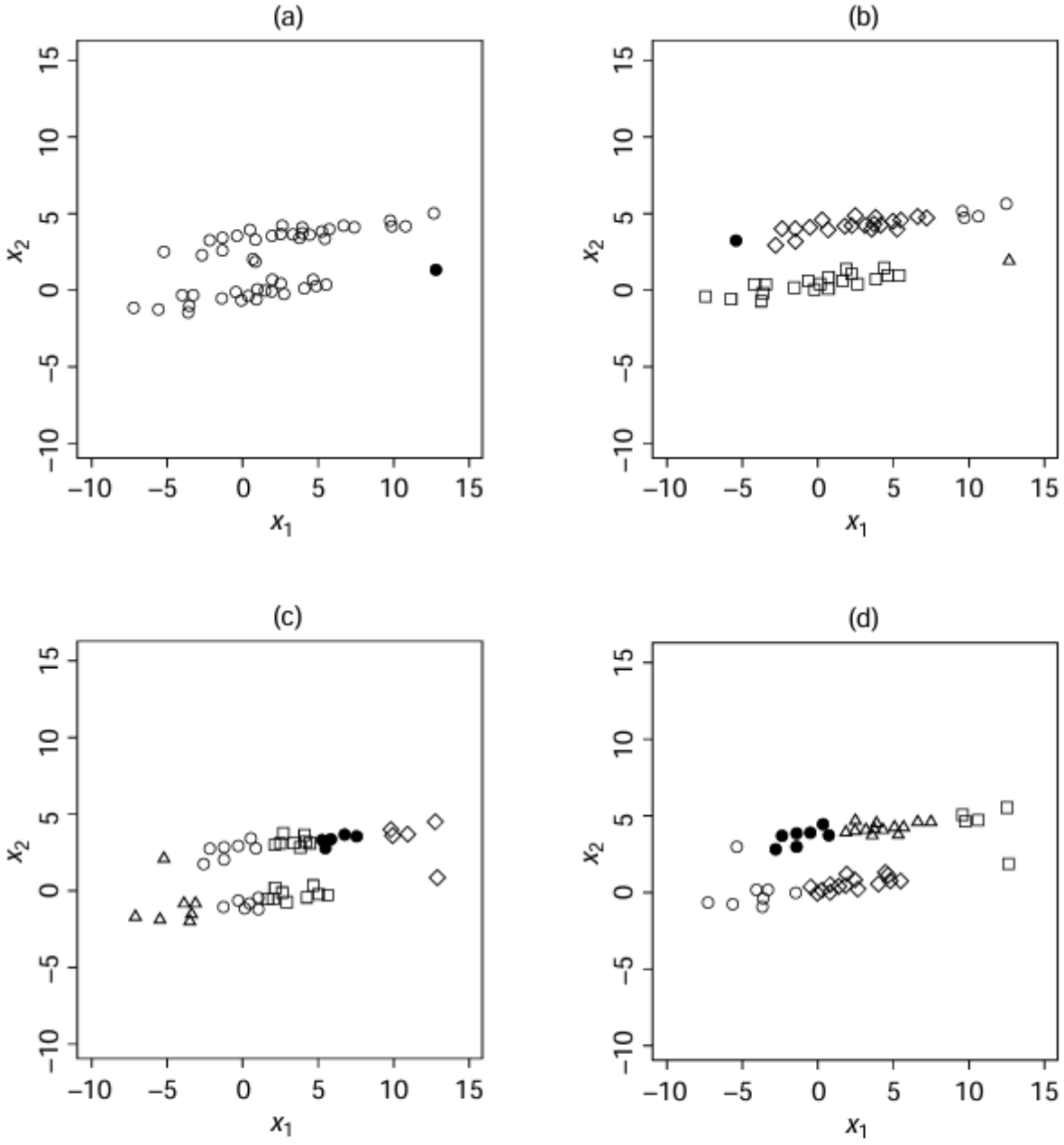


Figure 4.5 *Clusters obtained by four different methods from simulated data: (a) single linkage, with intermediate points, two-cluster solution; (b) single linkage, no intermediate points, five-cluster solution; (c) complete linkage, no intermediate points, five-cluster solution; (d) average linkage, no intermediate points, five-cluster solution.*

- These small examples show some of the problems of agglomerative methods and their relative failure to recover non-spherical clusters. (Similar problems were found in the more general empirical studies to be discussed in the next subsection.)
  - It also shows how crucial it is to make the correct choice as to the number of clusters present (see Section 4.4.4), and the advisability of plotting the raw data where feasible.
  - In the case of non uniqueness, a decision as to which clusters to fuse needs to be made, and this is usuallyadefaultchoicedeterminedbysoftware.Itisgenerallyrecommendedtorun analyses with different choices to check for robustness.

Empirical studies of hierarchical agglomerative methods

- Empirical studies of hierarchical methods are of two main types.
  - One type simulates clusters in data of a particular type and then assesses the characteristics and recovery of clusters.
    - Examples of the former include a review by Milligan (1981) and a study reported by Hands and Everitt (1987).
  - The other is based on real data from a particular subject matter, the criterion in the latter usually being the interpretability of clusters.
    - concluded that Ward's method performed very well when the data contained clusters with approximately the same numbers of points, but poorly when the clusters were of different sizes.
  - In that situation, centroid clustering appeared to give the most satisfactory results. Cunningham and Ogilvie (1972) and Blashfield (1976) also concluded that for clusters with equal numbers of points Ward's method is successful, otherwise centroid group average and complete linkage are preferable.
- Studies that focus on the stability of clustering in the presence of outliers or noise include that by Hubert (1974), who found that complete linkage is less sensitive to observational errors than single linkage.
  - A related point is the observation of Hartigan (1975), that single linkage is dependent on the smallest distances, and they need to be measured with low error for single linkage to be successful.
- An empirical study based on the subject-matter approach is that of Duflou and Maenhaut (1990). These authors compared seven standard methods (those in Table 4.1 and one other) on data involving chemical concentrations in the brain. They rejected centroid and median linkage because of reversals (a type of inconsistency in the hierarchy; see Section 4.4.3), and concluded that, of the remainder, Ward's method and complete linkage gave interpretable results and correctly distinguished grey and white matter areas in the brain.
  - A further example is provided by Baxter (1994), who summarizes the position in archaeology, where empirical studies generally favor Ward's method and average linkage.
- It has to be recognized that hierarchical clustering methods may give very different results on the same data, and empirical studies are rarely conclusive.
  - What is most clear is that no one method can be recommended above all others and, as Gordon (1998) points out, hierarchical methods are in any case only stepwise optimal. A few general observations can, however, be made.
  - Single linkage, which has satisfactory mathematical properties and is also easy to program and apply to large data sets, tends to be less satisfactory than other methods because of 'chaining'; this is the phenomenon in which separated clusters with 'noise' points in between them tend to be joined together. Ward's method often appears to work well but may impose a spherical structure where none exists.

Divisive methods

- Divisive methods operate in the opposite direction to agglomerative methods, starting with one large cluster and successively splitting clusters.

- They are computationally demanding if all $2^{k-1} - 1$ possible divisions into two subclusters of a cluster of *k* objects are considered at each stage.
    - However, for data consisting of p binary variables, relatively simple and computationally efficient methods, known as monothetic divisive methods, are available.
    - These generally divide clusters according to the presence or absence of each of the p variables, so that at each stage clusters contain members with certain attributes either all present or all absent. The data for these methods thus need to be in the form of a two-mode (binary) matrix.
- The term 'monothetic' refers to the use of a single variable on which to base the split at a given stage; polythetic methods, to be described in Section 4.3.2, use all the variables at each stage.
    - While less commonly used than agglomerative methods, divisive methods have the advantage, pointed out by Kaufman and Rousseeuw (1990), that most users are interested in the main structure in their data, and this is revealed from the outset of a divisive method.

Monothetic divisive methods

...

Applying the hierarchical clustering process

To make best use of hierarchical techniques, both agglomerative and divisive, the user often needs to consider the following points (in addition to the choice of proximity measure):

- graphical display of the clustering process
- comparison of dendrograms
- mathematical properties of methods
- choice of partition
- hierarchical algorithms

Dendrograms and other tree representations

- The dendrogram, or tree diagram, is a math and visual representation of the complete clustering procedure.

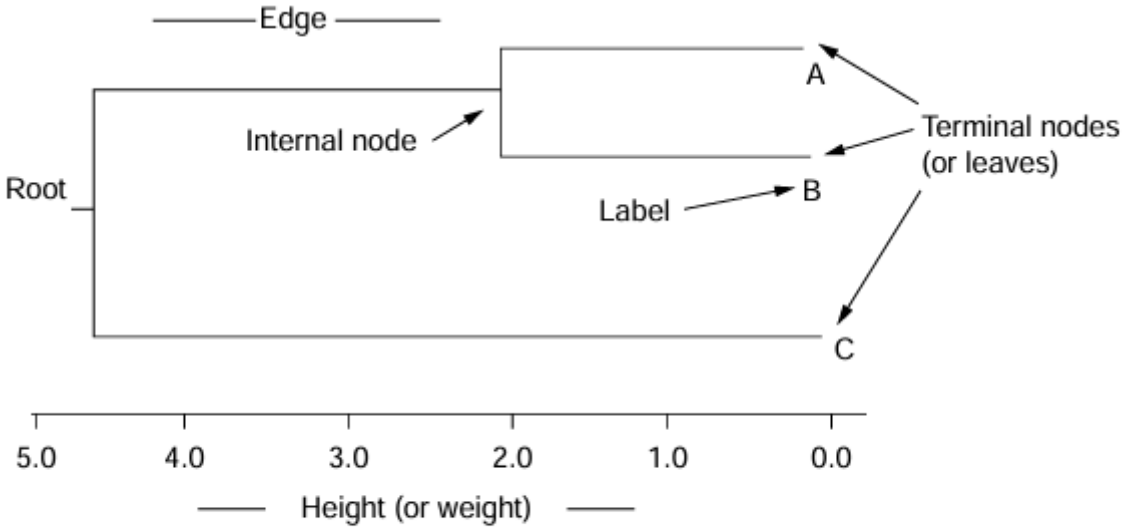


Figure 4.6 *Some terminology used in describing dendrograms.*

- Terminology
  - Nodes: represent clusters, and the length of the stems (*heights*) represent the distances at which clusters are joined.
    - Dendrograms which do not have numerical information attached to the stems are termed *unweighted* or *ranked*.
    - Binary trees: most dendrograms have two branches emanating from each node.
    - Topology: the arrangement of notes and stems of the tree.
  - Labels: the names of objects attached to the terminal nodes.
    - Internal nodes are not normally labelled. Typical or representative members of the clusters can be associated with the internal nodes, called *exemplars* or *centrotypes*, and are defined as the objects having maximum within-cluster average similarity.
    - A specific type of centrotype is the medoid (the object within the minimum absolute distance to the other members of the cluster).
  - Espaliers: generalized dendrogram, in which the length of the horizontal line conveys information about the relative homogeneity and separation of clusters.
  - The *pyramid* is a further specialized type of dendrogram for representing overlapping clusters.
- The *additive tree* (or *path length tree*) is a generalization of the dendrogram in which the lengths of the paths between the nodes represent proximities between objects, and in which the *additive inequality* (or *four-point condition*) holds.
  - This generalization of the ultrametric inequality is a necessary and sufficient condition for a set of proximities to be represented in the form of any additive tree.
  - The additive inequality is as follows:

$$d_{xy} + d_{uv} \leq \max |d_{xu} + d_{yu}| \text{ for all } x, y, u, v$$

  - Further details are given in Everitt and Rabe-Hesketh (1997), and an example of an additive tree showing genetic associations between various ethnic groups has kindly been provided by Kenneth Kidd (see Figure 4.7).
  - This is a representation of pairwise genetic distances among 30 human populations, generated by a searching routine (Kidd and Sgaramella-Zonta, 1971) that makes topological changes around

small or negative branches starting from the neighbour-joining tree produced by the PHYLIP package (Felsenstein, 1989).
- These branch lengths are the least-squares solution to the complete set of linear equations that relate each pairwise distance to the sum of the branch lengths connecting those populations.
- For these 30 populations there are $n(n-1)/2 = 435$ pairwise distances to be explained by addition of different combinations of $2n - 3 = 57$ branch lengths. Each tree topology is represented by a different set of equations. Of the 8.69×10^{36} possible trees (sets of linear equations), only about 100 were actually evaluated, and the tree in Figure 4.7 had the smallest Σe^2 (the quantity minimized by least squares for each set of linear equations); several others were almost as good as this tree, with only small differences around the very small branches.

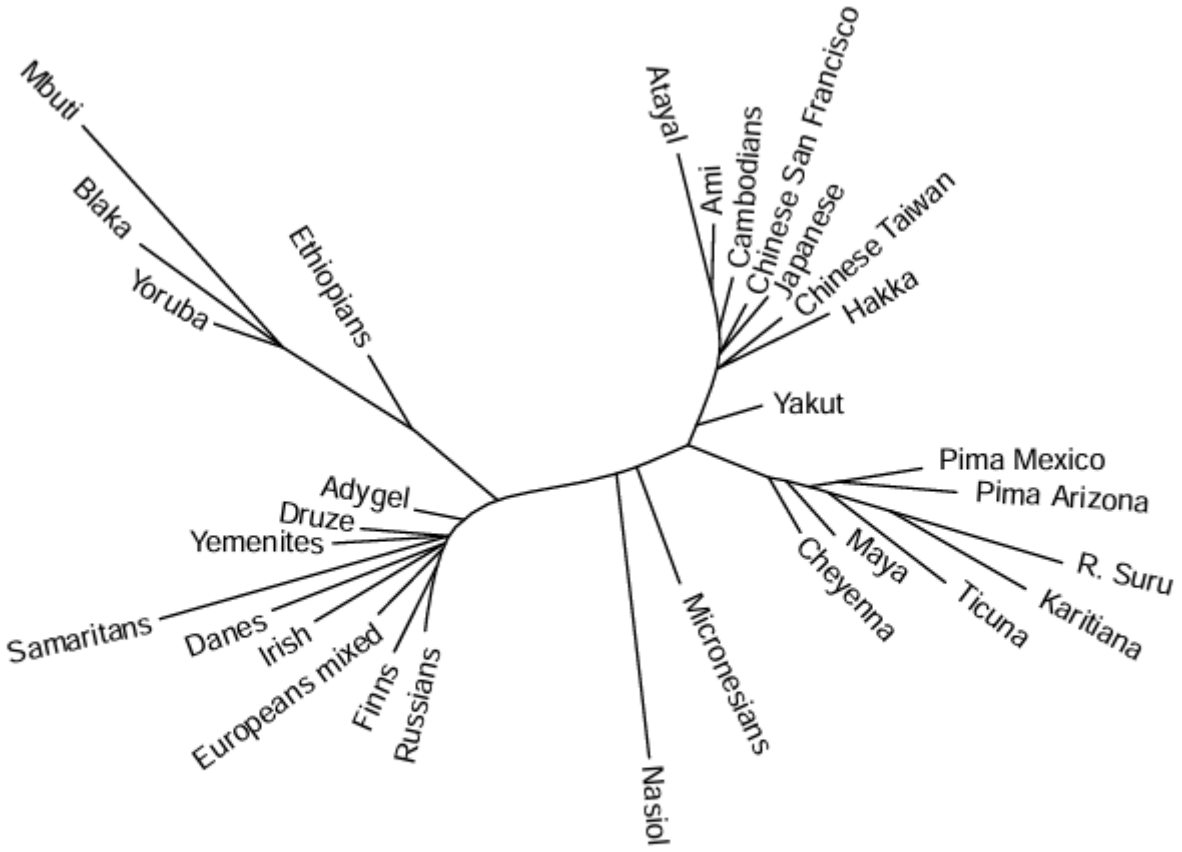


Figure 4.7 *An additive tree representation of pairwise genetic distances among 30 human populations; descriptions of these populations can be found in the ALFRED database at http://info.med.yale.edu/genetics/kkidd.* (Reproduced with permission from Kenneth K. Kidd.)

- It is important to realize that the same data and clustering procedure can give rise to 2^{n-1} dendrograms with different appearances, depending on the order in which the nodes are displayed.
  - This can be envisaged by imagining the dendrogram as a mobile in three-dimensional space: the stems from each node can swing around through 180 degrees without changing inter cluster relationships.
  - There are algorithms for optimizing the appearance of dendrograms. Look into Gale et al., 1984 and Degerman, 1982

Comparing dendrograms and measuring their distortion

- It may be required to compare two dendrograms without making a particular choice to the

particular partition corresponding to a specific number of clusters.

Mathematical properties of hierarchical methods

- Ultrametric Property:

$$h_{ij} \leq \max(h_{ij}, h_{jk}) \text{ for all } i, j, k$$

  - where h_{ij} is the distance between clusters i and j
  - For any three objects, the two largest distances between objects are equal. The property does not usually hold for the elements of proximity matrices. However, it does hold for heights h_{ij} at which two objects become members of the same cluster in many hierarchical clustering techniques.

Methods for large datasets

- For very large data sets where standard methods may not be able to cope, specialized methods have been developed. [3]
- Combining hierarchical method with a pre-clustering or sampling phase.
  - BIRCH: employs a pre clustering phase where dense regions are summarized, the summaries are then clustered using a hierarchical method based on centroids. [4]
  - CURE: starts with a random sample of points, and represents clusters by a smaller number of points that capture the shape of the cluster, which are then shrunk towards the centroid so as to dampen the effects of outliers; hierarchical clustering then operates on the representative points. [5]
    - CURE has been shown to be able to cope with arborary-shaped clusters, and in that respect may be superior to BIRCH, although it does require a judgment as to the number of clusters and also a parameter that favors more or less compact clusters.
  - SPSS Two-Step: the first step is similar to BIRCH in that it forms 'pre-clusters' by detecting dense regions; at this step, outliers (clusters with few changes) can be rejected before the next stage. The second step has some overlap with the model-based method described in chapter 6, in that one of the possible distances measures used in a combination of the likelihoods calculated for the continuous (assuming multivariate mixtures of normal distributions) and for the categorical variables (assuming multinomial distributions). [6]

Applications hierarchical methods

Dolphin whistles– agglomerative clustering

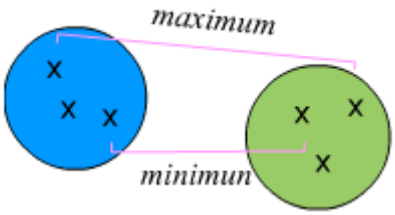
-

Agglomerative Clustering

the following are notes from Penn State's STAT897D-Applied Data Mining and Statistical Learning[7] course on Cluster Analysis[8]

- Agglomerative clustering can be used as long as we have pairwise distances between any two objects.

    - The mathematical representation of the objects are irrelevant when the pairwise distances are given. Hence agglomerative clustering readily applies for non-vector data.
- Let's denote the data set as $A = x_1, \ldots, x_n$.
- The agglomerative clustering method is also called a bottom-up method as opposed to k-means or k-center methods that are top-down. In a top-down method, a data set is divided into more and more clusters. In a bottom-up approach, all the data points are treated as individual clusters to start with and gradually merged into bigger and bigger clusters.
- In agglomerative clustering clusters are generated hierarchically. We start by taking every data point as a cluster. Then we merge two clusters at a time.
    - In order to decide which two clusters to merge, we compare the pairwise distances between any two clusters and pick a pair with the minimum distance.
    - Once we merge two clusters into a bigger one, a new cluster is created.
    - The distances between this new cluster and the existing ones are not given. Some scheme has to be used to obtain these distances based on the two merged clusters.
    - We call this the update of distances. Various schemes of updating distances will be described shortly.
- We can keep merging clusters until all the points are merged into one cluster. A tree can be used to visualize the merging process. This tree is called a dendrogram.
- The key technical detail here is how we define the between-cluster distance.
    - At the bottom level of the dendrogram where every cluster only contains a single point, the between-cluster distance is simply the Euclidian distance between the points. (In some applications, the between-point distances are pre-given.) However, once we merge two points into a cluster how do we get the distance between the new cluster and the existing clusters?
- The idea is to somehow aggregate the distances between the objects contained in the clusters.
    - For clusters containing only one data point, the between-cluster distance is the between-object distance.
    - For clusters containing multiple data points, the between-cluster distance is an agglomerative version of the between-object distances. There are a number of ways to accomplish this. Examples of these aggregated distances include the minimum or maximum between-object distances for pairs of objects across the two clusters.
- For example, suppose we have two clusters and each one has three points. One thing we could do is to find distances between these points. In this case, there would be nine distances between pairs of points across the two clusters. The minimum (or maximum, or average) of the nine distances can be used as the distance between the two clusters.



- How do we decide which aggregation scheme to use? Depending on how we update the distances, dramatically different results may come up. Therefore, it is always good practice to look at the results using scatterplots or other visualization methods instead of blindly taking the output of any algorithm. Clustering is inevitably subjective since there is no gold standard.

- Normally the agglomerative between-cluster distance can be computed recursively. The aggregation as explained above sounds computationally intensive and seemingly impractical. If we have two very large clusters, we have to check all the between-object distances for all pairs across the two clusters. The good news is that for many of these aggregated distances, we can update them recursively without checking all the pairs of objects. The update approach will be described soon.

  Example Distances
- Let's see how we would update between-cluster distances. Suppose we have two clusters r and s and these two clusters, not necessarily single point clusters, are merged into a new cluster t. Let k be any other existing cluster. We want the distance between the new cluster, t, and the existing cluster, k.
- We will get this distance based on the distance between k and the two component clusters, r and s.
- Because r and s have existed, the distance between r and k and the distance between s and k are already computed. Denote the distances by $D(r, k)$ and $D(s, k)$.
- We list below various ways to get $D(t, k)$ from $D(r, k)$ and $D(s, k)$.
- Single-link clustering:
  - $D(t, k) = min(D(r, k), D(s, k))$
  - $D(t, k)$ is the minimum distance between two objects in cluster t and k respectively. It can be shown that the above way of updating distance is equivalent to defining the between-cluster distance as the minimum distance between two objects across the two clusters.
- Complete-link clustering:
  - $D(t, k) = max(D(r, k), D(s, k))$
  - Here, $D(t, k)$ is the maximum distance between two objects in cluster t and k.
- Average linkage clustering:
  - There are two cases here, the unweighted case and the weighted case.
  - Unweighted case:

$$D(t, k) = \frac{n_r}{n_r + n_s} D(r, k) + \frac{n_s}{n_r + n_s} D(s, k)$$

    - Here we need to use the number of points in cluster *r* and the number of points in cluster *s* (the two clusters that are being merged together into a bigger cluster), and compute the percentage of points in the two component clusters with respect to the merged cluster. The two distances, $D(r, k)$ and $D(s, k)$, are aggregated by a weighted sum.
  - Weighted case:

$$D(t, k) = \frac{1}{2} D(r, k) + \frac{1}{2} D(s, k)$$

    - Instead of using the weight proportional to the cluster size, we use the arithmetic mean. While this might look more like an unweighted case, it is actually weighted in terms of the contribution from individual points in the two clusters. When the two clusters are weighted half and half, any point (i.e., object) in the smaller cluster individually contributes more to the aggregated distance than a point in the larger cluster. In contrast, if the larger cluster is given proportionally higher weight, any point in either cluster contributes equally to the aggregated distance.

- Centroid clustering:
  - A centroid is computed for each cluster and the distance between clusters is defined as the distance between their respective centroids.
  - Unweighted case:

$$D(t,k) = \frac{n_r}{n_r + n_s} D(r,k) + \frac{n_s}{n_r + n_s} D(s,k) - \frac{n_r n_s}{n_r + n_s} D(r,s)$$

  - Weighted case:

$$D(t,k) = \frac{1}{2} D(r,k) + \frac{1}{2} D(s,k) - \frac{1}{4} D(r,s)$$

- Ward's clustering:
  - We update the distance using the following formula:

$$D(t,k) = \frac{n_r + n_k}{n_r + n_s + n_k} D(r,k) + \frac{n_s + n_k}{n_r + n_s + n_k} D(s,k) - \frac{n_k}{n_r + n_s + n_k} D(r,s)$$

  - This approach attempts to merge the two clusters for which the change in the total variation is minimized. The total variation of a clustering result is defined as the sum of squared-errors between every object and the centroid of the cluster it belongs to.
  - The dendrogram generated by single-linkage clustering tends to look like a chain. Clusters generated by complete-linkage may not be well separated. Other methods are often intermediate between the two.

Application

Dataset Criteria:

- variables used in clustering should ideally not be interdependent
  Model Criteria:
- Cluster Groups can have varying sizes

Divisive methods

- Divisive methods operate in the opposite direction to agglomerative methods, starting with one large cluster and successively splitting clusters. They are computationally demanding if all 2k 1 1 possible divisions into two subclusters of a cluster of k objects are considered at each stage. However, for data consisting of p binary variables, relatively simple and computationally efficient methods, known as monothetic divisive methods, are available. These generally divide clusters according to the presence or absence of each of the p variables, so that at each stage clusters contain members with certain attributes either all present or all absent. The data for these methods thus need to be in the form of a two-mode (binary) matrix. The term ‘monothetic’ refers to the use of a single variable on which to base the split at a given stage; polythetic methods, to be described in Section 4.3.2, use all the variables at each stage. While less commonly used than agglomerative methods, divisive methods have the advantage, pointed out by Kaufman and Rousseeuw (1990), that most users are interested in the main structure in their data, and this is revealed from the outset of a divisive method.

It has the unique quality that it minimizes the sum of squares within (SSW) cluster groups whilst maximizing the sum of squares between (SSB) cluster groups.[9]

Applications in Studies

> In HA cluster analysis each measured particle is initially considered to represent its own single-membered cluster. The algorithm identifies two clusters with the highest degree of similarity, which are then agglomerated into a new cluster. This step is repeated until all particles populate a single cluster. The analyst is then required to determine which step (number of clusters) most appropriately represents the data, which is a subjective process, but may be informed by several statistics. There are several different HA cluster analysis algorithms, each defined by the respective metric used for comparing the similarity of clusters.

Cluster analysis of WIBS single-particle bioaerosol data, page 3

Hierarchical Cluster Analysis vs Bayesian statistics

Hierarchical Cluster Analysis (HCA) is not inherently a form of Bayesian Statistics. It is a machine learning and statistical clustering technique used to group data based on similarity, without requiring probabilistic assumptions about the data. However, Bayesian methods can be incorporated into clustering, leading to Bayesian hierarchical clustering, which differs from traditional HCA.

Key Differences Between Hierarchical Cluster Analysis and Bayesian Statistics

Aspect	Hierarchical Cluster Analysis (HCA)	Bayesian Statistics
Type of Method	Deterministic or distance-based clustering.	Probabilistic inference based on Bayes' theorem.
Underlying Principle	Measures similarity between data points using distance metrics (e.g., Euclidean, Manhattan).	Updates beliefs about a parameter using prior distributions and likelihoods.
Data Assumptions	Does not require explicit probability models.	Uses probability distributions for parameters.
Output	A dendrogram showing nested clusters.	Posterior distributions of model parameters.
Key Use Cases	Pattern recognition, taxonomy, and segmentation.	Bayesian inference, uncertainty quantification, and decision-making.

Bayesian Clustering: A Probabilistic Alternative

- Bayesian methods introduce prior probabilities and update cluster assignments based on data.
- Example: Bayesian Hierarchical Clustering (BHC):
  - Uses probability models for data partitions instead of a fixed distance metric.
  - Computes the posterior probability of cluster assignments using Bayes' theorem.

    - Allows for uncertainty quantification in cluster formation.
- Common models include:
    - Dirichlet Process Mixture Models (DPMM)
    - Bayesian Gaussian Mixture Models (BGMM)

When to Use Hierarchical Clustering vs. Bayesian Clustering

Scenario	HCA	Bayesian Clustering
No strong probabilistic assumptions	o	×
Large datasets with unknown structure	o	o
Need for probability-based clustering with uncertainty estimation	×	o
Computational efficiency (HCA is faster for small data)	o	×

Bayesian Hierarchical Clustering: How It Works

- BHC applies Bayesian probability to determine how likely it is that two data points belong to the same cluster. Instead of using predefined distance metrics, BHC computes the posterior probability of each cluster assignment, updating beliefs as new data is observed.
- Probabilistic Model for Clustering
    - Prior Distribution: Specifies the probability of a given clustering structure before seeing the data.
    - Likelihood Function: Describes how likely the observed data is given a particular clustering.
    - Posterior Probability: Updated belief about the clustering structure after incorporating observed data.
- Using Bayes' theorem, we compute:

$$P(C|D) = \frac{P(D|C)P(C)}{P(D)}$$

    - where,
        - $P(C|D) =$ Posterior probability of clustering C given data D
        - $P(D|C) =$ Likelihood (probability of observing data given clustering C)
        - $P(C) =$ Prior probability of clustering C
        - $P(D) =$ Normalizing factor (marginal likelihood)

Bayesian Models Used in Hierarchical Clustering

Several Bayesian nonparametric models are commonly used in BHC:

- Dirichlet Process Mixture Model (DPMM)
    - Allows an infinite number of clusters, adapting to data complexity.
    - Does not require a fixed number of clusters (K).
    - Clusters are formed probabilistically, balancing model fit and complexity.
    - Example Use Case: Clustering documents in natural language processing (NLP).

- Gaussian Mixture Model (GMM) with Bayesian Priors
  - Assumes clusters follow Gaussian distributions.
  - Uses Bayesian inference to estimate cluster parameters.
  - Handles uncertainty in cluster membership effectively.
  - Example Use Case: Identifying subtypes of cancer in genomic datasets.
- Chinese Restaurant Process (CRP)
  - Nonparametric clustering approach where each new data point joins an existing cluster with probability proportional to cluster size or starts a new cluster.
  - Highly flexible and adaptive to varying dataset structures.
  - Example Use Case: Clustering evolving user behavior patterns in recommendation systems.

Code

RStudio for hierarchical clustering
refer to Clustering Code - PennState > Hierarchical Clustering for a working R code

---

1. Cluster_Analysis_Chapter 4-hierarchical.pdf ↩
2. Cluster_Analysis_Chapter 3-measurment of proximity.pdf ↩ ↩ ↩ ↩
3. Zupan, J. (1982) Clustering of Large Data Sets . Research Studies Press, Chichester. (Everitt, Brian S., et al. *Cluster Analysis*, John Wiley & Sons, Incorporated, 2011. *ProQuest Ebook Central*, http://ebookcentral.proquest.com/lib/ttu/detail.action?docID=661789. Created from ttu on 2025-03-23 20:37:15.) ↩
4. Zhang, T., Ramakrishnan, R. and Livny, M. (1996) Birch: An efficient data clustering method for very large databases, in Proceedings of the ACM SIGMOD Conference on Management of Data , Montreal, Canada, 103–114. (Everitt, Brian S., et al. *Cluster Analysis*, John Wiley & Sons, Incorporated, 2011. *ProQuest Ebook Central*, http://ebookcentral.proquest.com/lib/ttu/detail.action?docID=661789. Created from ttu on 2025-03-23 20:36:47.) ↩
5. Guha, S., Rajeev, R. and Kyuseok, S. (1998) CURE: An efficient clustering algorithm for large databases, in Proceedings of the ACM SIGMOD Conference on Management of Data, Seattle, USA , 73– 84. (Everitt, Brian S., et al. *Cluster Analysis*, John Wiley & Sons, Incorporated, 2011. *ProQuest Ebook Central*, http://ebookcentral.proquest.com/lib/ttu/detail.action?docID=661789. Created from ttu on 2025-03-23 20:34:11.) (Cure: an efficient clustering algorithm for large databases - ScienceDirect) CURE- An efficient clustering algorithm for large databases, in Proceedings of the ACM SIGMOD Conference on Management of Data, Seattle, USA.pdf ↩
6. Chiu, T., Fang, D., Chen, J. et al. (2001) A robust and scalable clustering algorithm for mixed type attributes in large database environment, in KDD '01: Proceedings of the seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining . (Everitt, Brian S., et al. *Cluster Analysis*, John Wiley & Sons, Incorporated, 2011. *ProQuest Ebook Central*, http://ebookcentral.proquest.com/lib/ttu/detail.action?docID=661789. Created from ttu on 2025-03-23 20:35:33.) (A robust and scalable clustering algorithm for mixed type attributes in large database environment | Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining) A robust and scalable clustering algorithm for mixed type attributes in large database environment.pdf ↩
7. 12.6 - Agglomerative Clustering | STAT 897D ↩
8. Agglomerative Clustering-Penn State.pdf ↩
9. Cluster analysis of WIBS single-particle bioaerosol data.pdf ↩