CLUSTER ANALYSIS

Steven M. Holland

Department of Geology, University of Georgia, Athens, GA 30602-2501

January 2006

Introduction

Cluster analysis includes a broad suite of techniques designed to find groups of similar items within a data set. Partitioning methods divide the data set into a number of groups predesignated by the user. Hierarchical cluster methods produce a hierarchy of clusters from small clusters of very similar items to large clusters that include more dissimilar items. Hierarchical methods usually produce a graphical output known as a dendrogram or tree that shows this hierarchical clustering structure. Some hierarchical methods are divisive, that progressively divide the one large cluster comprising all of the data into two smaller clusters and repeat this process until all clusters have been divided. Other hierarchical methods are agglomerative and work in the opposite direction by first finding the clusters of the most similar items and progressively adding less similar items until all items have been included into a single large cluster. Cluster analysis can be run in the Q-mode in which clusters of samples are sought or in the R-mode, where clusters of variables are desired.

Hierarchical methods are particularly useful in that they are not limited to a pre-determined number of clusters and can display similarity of samples across a wide range of scales. Agglomerative hierarchical methods are particularly common in the natural sciences and they will be the focus of this lecture.

Computation

As with other multivariate methods, the starting point is a data matrix consisting of n rows of samples and p columns of variables, called an n x p (n by p) matrix. Hierarchical agglomerative cluster analysis begins by calculating a matrix of distances among items in this data matrix. Although cluster analysis can be run in the R-mode when seeking relationships among variables, this discussion will assume that a Q-mode analysis is being run. In Q-mode analysis, the distance matrix is a square, symmetric matrix of size n x n that expresses all possible pairwise distances among samples. Many distance metrics can be used.

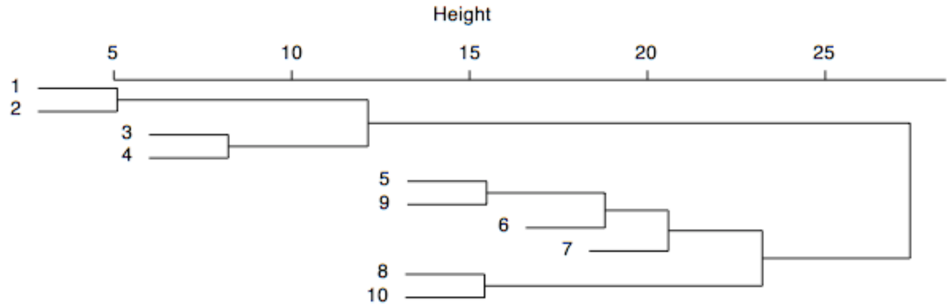
Before clustering has begun, each sample is considered a group, albeit of a single sample. Clustering begins by finding the two groups that are most similar, based on the distance matrix, and merging them into a single group. The characteristics of this new group are based on a combination of all the samples in that group. This procedure of combining two groups and merging their characteristics is repeated until all the samples have been joined into a single large cluster.

A variety of distance metrics can be used to calculate similarity. For data that show linear relationships, euclidean distance is a useful measure of distance. For data that show modal relationships, such as ecological data, the Sørenson distance is a better descriptor of similarity because it considers only taxa occurring in at least one of the samples.

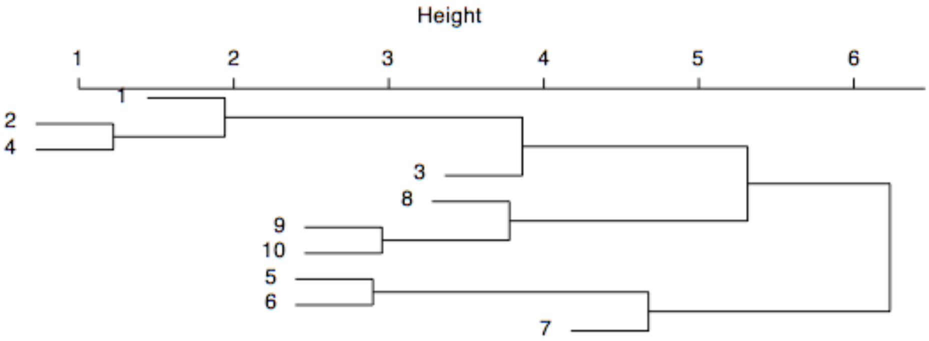
A variety of linkage methods can be used to determine in what order clusters may join. The nearest neighbor or single linkage method is based on the elements of two clusters that are most similar, whereas the farthest neighbor or complete linkage method is based on the elements that are most dissimilar. Both of these are based on outliers of distributions, which may not be desirable. The median, group average, and centroid methods all emphasize the central tendency of clusters and are less sensitive to outliers. Group average may be unweighted (also known as UPGMA) or weighted (WPGMA). Ward's method joins clusters based on minimizing the within-group sum of squares and will tend to produce compact clusters.

Interpretation of Dendrograms

The results of the cluster analysis are shown by a dendrogram, which lists all of the samples and indicates at what level of similarity any two clusters were joined. The x-axis is some measure of the similarity or distance at which clusters join and different programs use different measures on this axis. In the dendrogram shown above, samples 1 and 2 are the most similar and join to form the first cluster, followed by samples 3 and 4. The last two clusters to form are 1-2-3-4 and 5-9-6-7-8-10. Clusters may join pairwise, such as the joining of 1-2 and 3-4. Alternatively, individual samples may be sequentially added to an existing cluster, such as the join of 6 with 5-9, followed by the join of 7. Such sequential joining of individual samples is known as chaining.

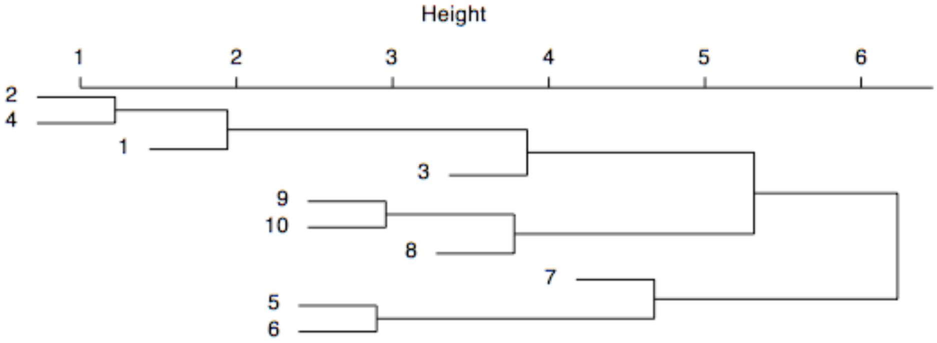


Determining the number of groups in a cluster analysis is often the primary goal. Although objective methods have been proposed, their application is somewhat arbitrary. Typically, one looks for natural groupings defined by long stems, such as the one to the right of cluster 1-2-3-4. Some have suggested that all clusters be defined at a consistent level of similarity, such that one would draw a line at some chosen level of similarity and all stems that intersect that line would indicate a group. The strength of clustering is indicated by the level of similarity at which elements join a cluster. In the example above, elements 1-2-3-4 join at similar levels, as do elements 5-9-6-7-8-10, suggesting the presence of two major clusters in this analysis. In contrast, the dendrogram below displays more chaining, clustering at a wider variety of levels,



and a lack of long stems, suggesting that there are no discrete clusters within the data. Defining groups involves a tradeoff between the number of groups and the similarity of elements in the group. If many groups are defined, they will be small in size and their elements will be highly similar, but the analysis of a great many groups can be difficult. If fewer groups are defined, their larger number of elements will show less similarity to one another, but the smaller number of groups will be easier to analyze.

Dendrograms are like mobiles and can be freely rotated around any node. In the example above, the dendrogram could be spun such that the samples appeared in a different order, shown below, with the constraint that the dendrogram does not cross itself. Such spinning of a dendrogram is a useful way to accentuate patterns of chaining or the distinctiveness of clusters (although it doesn't aid in this case). For many cluster analysis programs, spinning must be done (tediously) in a separate graphics program, such as Illustrator, but spinning can be done much more easily directly in R.



Considerations

Because of its agglomerative nature, clusters are sensitive to the order in which samples join, which can cause samples to join a grouping to which it does not actually belong. In other words, if groups are known beforehand, those same groupings may not be produced from cluster analysis.

Cluster analysis is sensitive to both the distance metric selected and the criterion for determining the order of clustering. Different approaches may yield different results. Consequently, the distance metric and clustering criterion should be chosen carefully. The results should also be compared to analyses based on different metrics and clustering criteria, or to an ordination, to determine the robustness of the results.

Caution should be used when defining groups based on cluster analysis, particularly if long stems are not present. Even if the data form a cloud in multivariate space, cluster analysis will still form clusters, although they may not be meaningful or natural groups. Again, it is generally wise to compare a cluster analysis to an ordination to evaluate the distinctness of the groups in multivariate space.

Transformations may be needed to put samples and variables on comparable scales; otherwise, clustering may reflect sample size or be dominated by variables with large values.

Cluster Analysis in R

The cluster package in R includes a wide spectrum of methods, corresponding to those presented in Kaufman and Rousseeuw (1990). Curiously, the methods all have the names of women that are derived from the names of the methods themselves. Of the partitioning methods, pam is based on partitioning around mediods, clara is for clustering large applications, and fanny uses fuzzy analysis clustering. Of the hierarchical methods, agnes uses agglomerative nesting, diana is based on divisive analysis, and mona is based on monothetic analysis of binary variables. Other functions include daisy, which calculates dissimilarity matrices, but is limited to euclidean and manhattan distance measures. The agnes method will be the focus of this tutorial.

1) Load the cluster library needed to run agnes() and other clustering functions.

```
> library(cluster)
```

2) Run a default cluster analysis using agnes(), without prior transformations of the data. The defaults include a euclidean distance measure, an average (UPGMA) linkage method, and no standardization of variables.

```
> mydata <- read.table(file="mydata.txt", header=TRUE,
   row.names=1, sep=",")
> mydata.agnes <- agnes(mydata)
```

3) Customize the cluster analysis by changing the distance measure (metric), the linkage method (method), and standardization of variables (stand). Standardization transforms the observations for each variable to have a zero mean and a unit variance, to prevent particular variables from dominating the analysis. Distance metrics are limited in agnes() to euclidean and manhattan.

```
> mydata.agnes.ALT <- agnes(mydata, metric="manhattan",
   method="ward", stand=TRUE)
```

4) Run the cluster analysis on a distance matrix rather than a data matrix. This will allow the use of a different distance metric (e.g., Sørenson), but after the initial cluster is formed, all subsequent calculations will use either the manhattan or euclidean measure, depending on how agnes() is called.

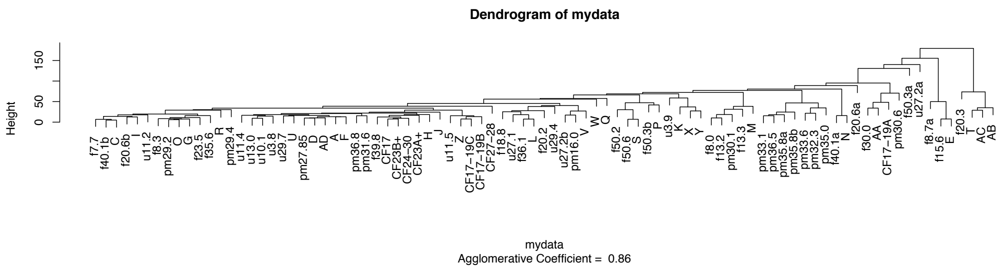
```
> library(vegan)
# load library for distance functions

> mydata.bray <- vegdist(mydata, method="bray")
# calculates bray (=Sørenson) distances among samples

> mydata.bray.agnes <- agnes(mydata.bray)
# run the cluster analysis
```

5) View items in list produced by agnes.

```
> names(mydata.agnes)
# mydata.agnes$order: numeric order of the samples on the
#  dendrogram, with 1 being the sample at the left/top of the
#  dendrogram
# mydata.agnes$height: stores distances of merging clusters
# mydata.agnes$ac: agglomerative coefficient, a measure of the
#  clustering structure, but is sensitive to sample size
# mydata.agnes$merge: describes merging of clusters at each
#  step
# mydata.agnes$diss: dissimilarity matrix of dataset,
#  including dissimilarity measure and number of objects
# mydata.agnes$call: how the function was called
# mydata.agnes$method: linkage method
# mydata.agnes$order.lab: like order, but with sample labels
# mydata.agnes$data: matrix of data or dissimilarities named
#  in call to function
```

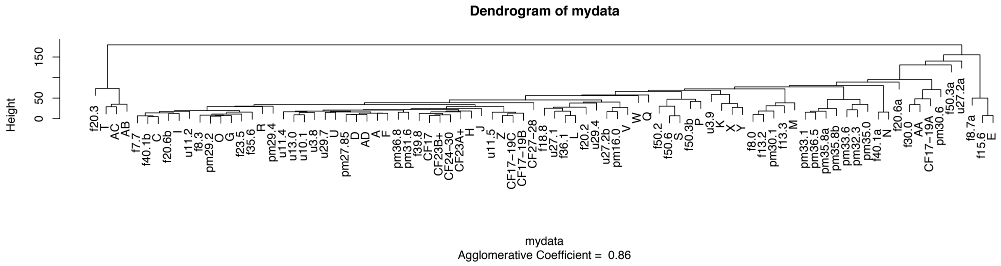
6) Plot the dendrogram.

```
> plot(mydata.agnes, which.plots = 2, main=”Dendrogram of my
  data”)
# which.plots=2 specifies the dendrogram
```



7) Spin the dendrogram. By editing the mydata.agnes$order and mydata.agnes&order.lab elements, it is possible to specify the left-to-right positions of the sample in the dendrogram. This must be done carefully to avoid a dendrogram that crosses over itself. In this case, we will spin the four rightmost samples to the left side of the dendrogram. It is best to work on copies of order and order.lab, in case mistakes are made.

```
> mydata.agnes.order <- mydata.agnes$order
> mydata.agnes.order.lab <- mydata.agnes$order.lab
# these are backups, in case something goes wrong

> working.order <- mydata.agnes$order
> working.order.lab <- mydata.agnes$order.lab
# these will be the copies that will be modified

> working.order <- c(working.order[84:87],
  working.order[1:83])
> working.order.lab <- c(working.order.lab[84:87],
  working.order.lab[1:83])
# reordering the samples and labels

> mydata.agnes$order <- working.order
> mydata.agnes$order.lab <- working.order.lab
# overwriting the original order and labels

> plot(mydata.agnes, which.plots = 2, main="Dendrogram of my
  data")
```



References

Kaufman, L. and P.J. Rousseeuw, 1990. Finding Groups in Data: An Introduction to Cluster Analysis. Wiley, New York.

Legendre, P., and L. Legendre, 1998. Numerical Ecology. Elsevier: Amsterdam, 853 p.

McCune, B., and J.B. Grace, 2002. Analysis of Ecological Communities. MjM Software Design: Gleneden Beach, Oregon, 300 p.