Chapter 6: Cluster Analysis

- The major goal of cluster analysis is to separate individual observations, or *items*, into groups, or *clusters*, on the basis of the values for the q variables measured on each individual.

- Often in clustering the items are called *objects*.

- We wish to create clusters such that the objects within each cluster are *similar* and objects in different clusters are dissimilar.

- The dissimilarity between any two objects is typically quantified using a distance measure (like Euclidean distance).

- Cluster analysis is a type of *unsupervised classification*, because we do not know the nature of the groups (or the number of groups, typically) before we classify the objects into clusters.

Applications of Cluster Analysis

- In marketing, researchers attempt to find distinct clusters of the consumer population so that several distinct marketing strategies can be used for the clusters.
- In ecology, scientists classify plants or animals into various groups based on some measurable characteristics.
- Researchers in genetics may separate genes into several classes based on their expression ratios measured at different time points.

The Need for Clustering Algorithms

- We assume there are n objects that we wish we separate into a small number (say, k) of clusters, where $k < n$.
- If we know the true number of clusters k ahead of time, then the number of ways to partition the n objects into k clusters is a “Stirling number of the second kind.”
- *Example*: There are $48,004,081,105,038,305$ ways to separate $n = 30$ objects into $k = 4$ clusters.

The Need for Clustering Algorithms (Continued)

- If we don't know the value of k, the possible number of partitions is even more massive.

- *Example*: There are $35,742,549,198,872,617,291,353,508,656,626,642,567$ possible partitions of $n = 42$ objects if we let the number of clusters k vary. (This is called the *Bell number* for n.)

- Clearly even a computer cannot investigate all the possible clustering partitions to see which is best.

- We need intelligently designed *algorithms* that will search among the best possible partitions relatively quickly.

Types of Clustering Algorithms

- There are three major classes of clustering methods – from oldest to newest, they are:

  1. Hierarchical methods
  2. Partitioning methods
  3. Model-based methods

- Hierarchical methods cluster the data in a series of n steps, typically joining observations together step by step to form clusters.

- Partitioning methods first determine k, and then typically attempt to find the partition into k clusters that optimizes some *objective function* of the data.

- Model-based clustering takes a statistical approach, formulating a model that categorizes the data into subpopulations and using maximum likelihood to estimate the model parameters.

Hierarchical Clustering

- *Agglomerative* hierarchical clustering begins with n clusters, each containing a single object.
- At each step, the two clusters that are “closest” are merged together.
- So as the steps iterate, there are n clusters, then $n-1$ clusters, then $n-2$, etc.
- By the last step, there is 1 cluster containing all n objects.
- The `R` function `hclust` will perform a variety of hierarchical clustering methods.

Defining Closeness of Clusters

- The key in a hierarchical clustering algorithm is specifying how to determine the two “closest” clusters at any given step.
- For the first step, it’s easy: Join the two objects whose (Euclidean?) distance is smallest.
- After that, we have a choice: Do we join two individual objects together, or merge an object into a cluster that already has multiple objects?
- Intercluster dissimilarity is typically defined in one of three ways, which give rise to *linkage methods*.

Linkage Methods in Hierarchical Clustering

- The *single linkage* algorithm, at each step, joins the clusters whose *minimum* distance between objects is smallest, i.e., joins the clusters A and B with the smallest

$$d_{AB} = \min_{i \in A, j \in B} (d_{ij})$$

- *Single linkage* clustering is sometimes called “nearest neighbor” clustering.
- The *complete linkage* algorithm, at each step, joins the clusters whose *maximum* distance between objects is smallest, i.e., joins the clusters A and B with the smallest

$$d_{AB} = \max_{i \in A, j \in B} (d_{ij})$$

- *Complete linkage* clustering is sometimes called “farthest neighbor” clustering.

Linkage Methods in Hierarchical Clustering (Continued)

- The *average linkage* algorithm, at each step, joins the clusters whose *average* distance between objects is smallest, i.e., joins the clusters A and B with the smallest

  $$d_{AB} = \frac{1}{n_A n_B} \sum_{i \in A} \sum_{j \in B} d_{ij}$$

  where n_A and n_B are the number of objects in clusters A and B, respectively.

- See Figure 6.4 in the Everitt textbook.

Dendrograms

- A hierarchical algorithm actually produces not one partition of the data, but lots of partitions.
- There is a clustering partition for each step $1, 2, \ldots, n$.
- The series of mergings can be represented at a glance by a treelike structure called a *dendrogram*.
- To get a single k-cluster solution, we would cut the dendrogram horizontally at a point that would produce k groups (The R function `cutree` can do this).
- It is strongly recommended to examine the full dendrogram first before determining where to cut it.
- A natural set of clusters may be apparent from a glance at the dendrogram.

Standardization of Observations

- If the variables in our data set are of different types or are measured on very different scales, then some variables may play an inappropriately dominant role in the clustering process.

- In this case, it is recommended to standardize the variables in some way before clustering the objects.

- Possible standardization approaches:

1. Divide each column by its sample standard deviation, so that all variables have standard deviation 1.
2. Divide each variable by its sample range ($\max - \min$); Milligan and Cooper (1988) found that this approach best preserved the clustering structure.
3. Convert data to z-scores by (for each variable) subtracting the sample mean and then dividing by the sample standard deviation – a common option in clustering software packages.

Pros and Cons of Hierarchical Clustering

- An advantage of hierarchical clustering methods is their computational speed for small data sets.

- Another advantage is that the dendrogram gives a picture of the clustering solution for a variety of choices of k (the number of clusters) at once.

- On the other hand, a major disadvantage is that once two clusters have been joined, they can never be split apart later in the algorithm, even if such a move would improve the clustering.

- The so-called *partitioning methods* of cluster analysis do not have this restriction.

- In addition, hierarchical methods can be less efficient than partitioning methods for large data sets, when n is much greater than k.