Clustering Methods

Clustering is a form of unsupervised learning in which we try to divide the dataset into clusters (groups) of cases so that cases within a group are similar, while cases in different clusters are dissimilar.

To do this, we first need to have a measure of "dissimilarity".

We can then find a "flat" clustering of the data by choosing some number of clusters, K, and then either

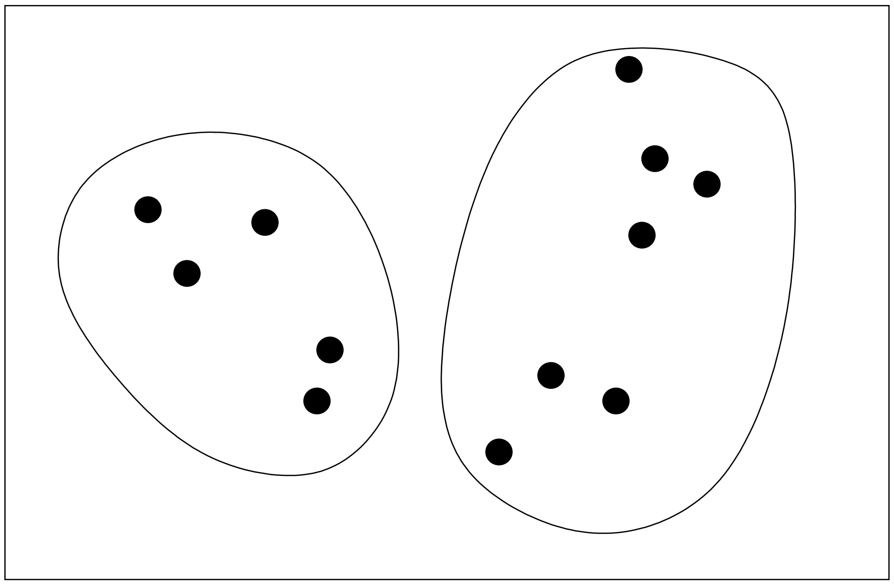
- Try to assign cases to clusters to directly minimize the "within cluster" scatter. Doing this exactly is possible only for small datasets.
- Make some initial assignment of cases to clusters, and then apply an iterative algorithm to try to improve the clustering.

An alternative is to find a *hierarchical* clustering (a tree), by either

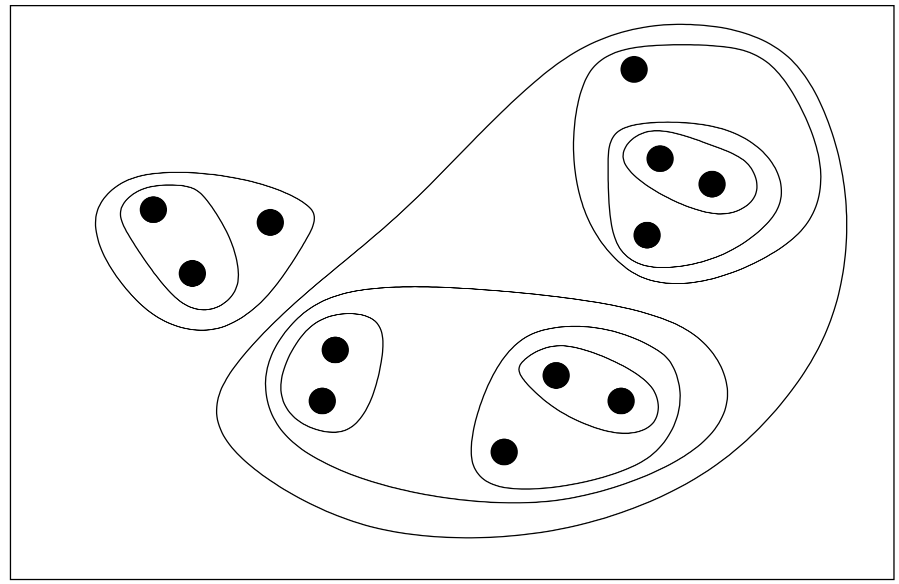
- Beginning with all cases in one cluster, and then iteratively dividing clusters until every case is in its own cluster (*divisive* clustering).
- Begin with a cluster for every case, and then iteratively merge clusters until there is just one cluster (*agglomerative* clustering).

Examples of Flat and Hierarchical Clusterings

Here are two clusterings of a 2D dataset with 12 cases, with dissimilarity measured by Euclidean distance.



A flat clustering with two clusters



A hierarchical clustering

From the hierarchical clustering, we can obtain many flat clusterings by eliminating parts of the clustering above or below some level.

Dissimilarity

The dissimilarities of N cases can be expressed as an $N \times N$ matrix, $\mathbf{D}$, with elements $d_{ii'}$. We require that $\mathbf{D}$ be symmetric, and that the diagonal elements, d_{ii}, be zero.

We might fill in this matrix directly — eg, ask someone to rate how similar each pair of cases is.

Instead, if we have measurements on p variables for each case, we can define dissimilarity by the sum of dissimilarities between values of these variables:

$$d_{ii'} \;=\; D(x_i, x_{i'}) \;=\; \sum_{j=1}^{p} d_j(x_{ij}, x_{i'j})$$

where d_j measures the dissimilarity of two values for variable j.

If variable j is real-valued, the common choice is $d_j(x_{ij}, x_{i'j}) = (x_{ij} - x_{i'j})^2$, which makes dissimilarities be squared Euclidean distances.

For a categorical variable, we could let $d_j(x_{ij}, x_{i'j})$ be zero if the $x_{ij} = x_{i'j}$, and one otherwise.

Within Cluster and Between Cluster Scatter

The total "scatter" of the cases is the sum of dissimilarities for all pairs of points, which can be written as

$$T = \frac{1}{2} \sum_{i=1}^{N} \sum_{i'=1}^{N} d_{ii'}$$

For a given clustering, C, with K clusters, the "within cluster" scatter is

$$W(C) = \frac{1}{2} \sum_{k=1}^{K} \sum_{i\,:\,C(i)=k} \; \sum_{i'\,:\,C(i')=k} d_{ii'}$$

Here, $C(i)$ is the cluster (represented by a number from 1 to K) that case i is assigned to according to the clustering C.

The "between cluster" scatter is

$$B(C) = \frac{1}{2} \sum_{k=1}^{K} \sum_{i\,:\,C(i)=k} \; \sum_{i'\,:\,C(i')\neq k} d_{ii'}$$

It's easy to see that $T = W(C) + B(C)$.

Clustering by Minimizing Within Cluster Scatter

Suppose we have chosen the number of clusters, K, that we want. We can try to find the clustering, C, with K clusters that minimizes $W(C)$, or equivalently, that maximizes $B(C)$.

If N is at all large, however, there are an enormous number of possible clusterings with K clusters. According to the book, if $K = 4$, there are 34,105 clusterings when $N = 10$ and 10^{10} clusterings with $N = 19$.

If we're clever, we don't have to look at all of these to find the optimal one, but it's still infeasible to find the optimal clustering when N is large.

Instead, we can start with some initial clustering, and try to make small changes to improve it. We stop when none of the changes are an improvement. This gets us a local optimum, but not necessarily the best clustering.

K-Means Clustering

When the dissimilarities are squared Euclidean distances, we can use the *K-means* algorithm to iteratively find a clustering with K clusters.

We start with some initial clustering — perhaps just a random division of the cases into K clusters.

We then repeatedly apply the following two steps to improve the clustering:

1) Using the current clustering, C, find the mean vectors for each cluster:

$$m_k \;=\; \frac{1}{\#\{i\,:\,C(i)=k\}} \sum_{i\,:\,C(i)=k} x_i$$

2) Update the clustering by assigning each case to the cluster with the closest mean. Ie, the new clustering, C, is defined by

$$C(i) \;=\; \underset{1\leq k\leq K}{\operatorname{argmin}}\, ||x_i - m_k||^2$$

We stop when step (2) doesn't change C. Figure 14.6 shows an example.

This algorithm finds a local minimum of the within-cluster scatter. It's advisable to run it several times with different initial clusterings, then use the best result.

Hierarchical Clustering

Deciding on the number of clusters for a flat cluster is a problem. We can avoid it by finding a *hierarchical* clustering, in which we have clusters of clusters, in a tree.

Looking at different levels in the tree gives clusterings at various levels of detail. We might pick one such flat clustering, or look at the whole hierarchy.

Hierarchies are common ways of organizing categories. Biological taxonomies (species, genera, families, etc.) are elaborate hierarchies.

There are two sorts of algorithms for finding hierarchical clusterings:

- *Divisive* algorithms start with everything in one cluster, and then divide clusters until everything is in its own cluster.

- *Agglomerative* algorithms start with everything in its own cluster, and then merge clusters until everything is in one cluster.

Deciding what clusters to merge requires that we extend our dissimilarity measure for cases to a dissimilarity measure for clusters. This can be done in various ways, giving various agglomerative clustering methods.

Average Linkage Agglomerative Clustering

The most common way of measuring dissimilarity between clusters is to just average the dissimilarities for all pairs with one case in each cluster — that is, the dissimilarity of clusters G and H, containing N_G and N_H cases, is

$$\frac{1}{N_G N_H} \sum_{i \in G} \sum_{i' \in H} d_{ii'}$$

Using this gives the "average linkage" hierarchical clustering algorithm:

1) Assign every case to its own cluster.
2) Repeat the following, until everything is in one cluster:
   a) Find the two clusters with smallest dissimilarity.
   b) Replace these two clusters with one cluster, containing the cases in both of the merged clusters.

We keep track of all the clusters formed in step (2a). They form a tree, in which each merged cluster is the "parent" of the two clusters that were merged.

A picture of this tree is called a *dendrogram*. Sometimes the lengths of the branches in the dendrogram represent how far apart the merged clusters were.