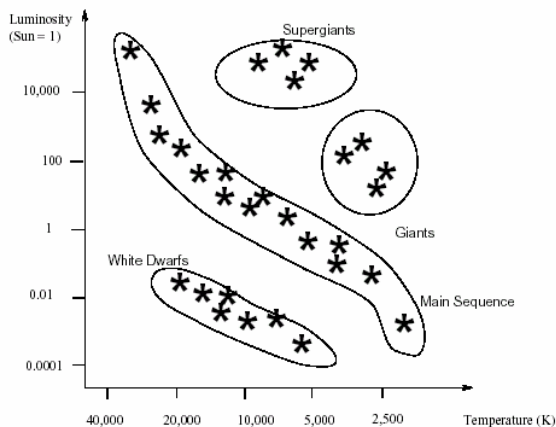


Clustering

Cluster analysis is used to group “similar” records, where similarity is measured using a *distance measure*. This is quite a different problem to classification, where we are told which groups records belong to. Here we know nothing a-priori about how many groups there are or what shape they have. Clustering is known as *unsupervised learning* in the machine learning community.

The Hertzsprung-Russell Diagram



(Berry/Linoff, 1997)

Hierarchical methods

Suppose we have N records $\mathbf{x}(1), \dots, \mathbf{x}(N)$, where $\mathbf{x}(i) = (x_1(i), \dots, x_n(i))$. Define the distance or *dissimilarity matrix* by

$$D = \begin{pmatrix} 0 & d_{1,2} & d_{1,3} & \dots & d_{1,n} \\ d_{2,1} & 0 & d_{2,3} & \dots & d_{2,n} \\ \vdots & & & & \vdots \\ d_{n,1} & d_{n,2} & d_{n,3} & \dots & d_{n,n} \end{pmatrix}$$

where d_{ij} is the distance between $\mathbf{x}(i)$ and $\mathbf{x}(j)$. We will assume that D is symmetric ($d_{ij} = d_{ji}$) and that the triangle inequality is satisfied ($d_{i,k} \leq d_{i,j} + d_{j,k}$).

Some possible definitions of d_{ij} are

Euclidean distance: $d_{i,j} = \left(\sum_{k=1}^n (x_k(i) - x_k(j))^2 \right)^{1/2}$.

Pearson distance: $d_{i,j} = \left(\sum_{k=1}^n (x_k(i) - x_k(j))^2 / S_k^2 \right)^{1/2}$ where $S_k^2 = \sum_{i=1}^N (x_k(i) - \bar{x}_k)^2 / (N-1)$ and $\bar{x}_k = \sum_{i=1}^N x_k(i) / N$. This is equivalent to using the Euclidean distance with standardised data.

Mahalanobi's distance: $d_{i,j}^2 = (\mathbf{x}(i) - \mathbf{x}(j))V^{-1}(\mathbf{x}(i) - \mathbf{x}(j))'$ where V is the sample covariance matrix.

City block or Manhattan metric: $d_{i,j} = \sum_{k=1}^n |x_k(i) - x_k(j)|$.

L_∞ norm: $d_{i,j} = \sup_k |x_k(i) - x_k(j)|$.

Minkowski metric: $d_{i,j} = \left(\sum_{k=1}^n |x_k(i) - x_k(j)|^p\right)^{1/p}$. This generalises the Euclidean and Manhattan distances. As $p \rightarrow \infty$ it converges to the L_∞ norm.

Discrete metric: $d_{i,j} = \sum_{k=1}^n I(x_k(i) \neq x_k(j))$. Particularly useful for categorical data.

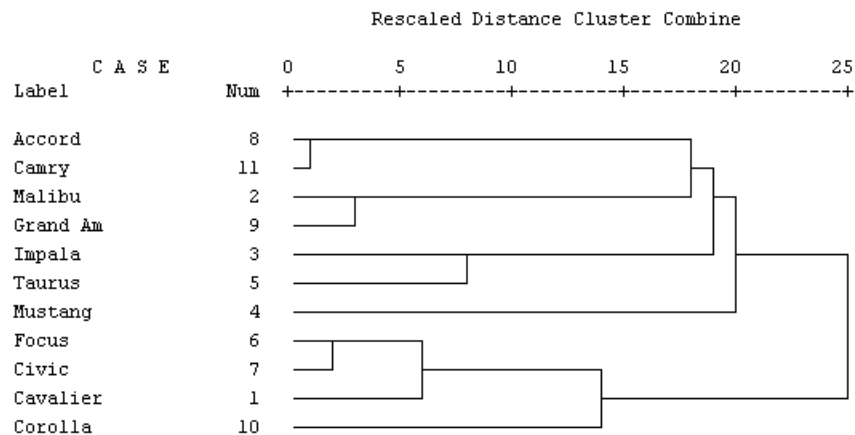
Different distances can be applied to different parts of the record vectors, in particular this may be necessary if some of the data is categorical and some not.

The range of values in any one co-ordinate/variable will affect the importance of that co-ordinate on the distance measure, and thus its importance when forming clusters. This can be avoided by scaling within the metric (as with the Pearson distance) or by explicitly scaling the data before hand. Alternatively, if one co-ordinate is known to be more important, it can be scaled to increase its influence.

Agglomeration

Clusters are formed from the bottom up. We start with N clusters each consisting of a single record, and then iteratively join the two "closest" clusters until we have one big cluster. This produces a binary tree of clusters, called a *dendrogram*. For example

Dendrogram using Single Linkage



The point at which a split occurs represents the distance between the two clusters.

There are several ways to define the distance between two clusters A and B. We call these *linkage methods*

Single linkage: $d(A, B) = \min\{d_{i,j} : i \in A, j \in B\}$

Complete linkage: $d(A, B) = \max\{d_{i,j} : i \in A, j \in B\}$

Average linkage: $d(A, B) = \text{mean}\{d_{i,j} : i \in A, j \in B\}$

Centroid linkage: $d(A, B) = d(a, b)$ where a and b are the *centroids* of A and B . The centroid is the arithmetic mean of all points in the cluster, taken component-wise. Note the difference between this and the average linkage, which is an average of distances rather than a distance of averages.

You can check that all of these linkage methods have the desirable property that

$$d(A, B) < \min\{d(A, C), d(B, C)\} \Rightarrow d(A, B) < d(A \cup B, C)$$

Note that each linkage method will work better with some distance measures than with others.

Ward's method and sums of squares

If we are using the Euclidean distance, then an alternative approach to agglomeration is Ward's method. We calculate the deviance within groups and the deviance between groups, and when combining groups we prefer agglomerations that give a big increase in the deviance between groups while giving only a small increase in the deviance within groups. Let \bar{x}_k be the mean of variable k and $\bar{x}_k(i)$ the mean of variable k in group i , then the deviance within groups is

$$W = \sum_{\text{groups } i} \sum_{j \text{ in group } i} \sum_{k=1}^n (x_k(j) - \bar{x}_k(i))^2 = \sum_{\text{groups } i} \sum_{j \text{ in group } i} \|x(j) - \bar{x}(i)\|^2$$

and the deviance between groups is

$$B = \sum_{\text{groups } i} N(i) \sum_{k=1}^n (\bar{x}_k(i) - \bar{x}_k)^2 = \sum_{\text{groups } i} N(i) \|\bar{x}(i) - \bar{x}\|^2$$

where $N(i)$ is the size of group i . The sum of B and W is the total sum of squares:

$$\begin{aligned} B + W &= \sum_{j=1}^N \sum_{k=1}^n (x_k(j) - \bar{x}_k)^2 = \sum_{j=1}^N \|x(j) - \bar{x}\|^2 \\ &= \frac{1}{2} \sum_{\text{groups } i} \sum_{\text{groups } i'} \sum_{\substack{j \in i \\ j' \in i'}} \|x(j) - x(j')\|^2 \end{aligned}$$

Performance measures

In practice when clustering one should try a variety of distances and linkages and compare the results. Groupings which appear consistently can be taken as more reliable.

A quantitative measure of how good a clustering is, is given by

$$R^2 = \frac{B}{B + W}$$

For a given number of clusters, the larger R^2 is the better. Note that R^2 will always increase as the number of clusters increases. None-the-less it can also be used to judge the optimal number of groups. If we plot R^2 against the number of clusters there is often a plateau; the start of the plateau is a good

indication of the optimal number of groups. In the dendrogram this corresponds to a big jump between splits.

Other performance measures include the pseudo-F criterion, the root mean square standard deviation (RMSSTD) and the semipartial R^2 .

Partitioning methods

Hierarchical methods are necessarily slow, as $O(N^2)$ calculations required to calculate D . Also the order in which clusters appear can be very data dependent. Partitioning methods are generally better on both of these fronts, but require the number of clusters to be specified before hand. In practice this is dealt with by trying several possibilities and using a performance measure to compare the results.

k-means algorithm

The *k*-means algorithm uses the Euclidean distance to split the observations into *k* clusters.

Let $c(j) = i$ if $x(j)$ is assigned to cluster i . The size of cluster i will be denoted $N(i) = |\{j : c(j) = i\}|$.

For a given assignment of observations to clusters, our measure of the quality of the assignment is the within groups sum of squares, that is

$$W = \sum_{i=1}^k \sum_{j:c(j)=i} \|x(j) - \bar{x}(i)\|^2,$$

where $\bar{x}(i)$ is the centroid of cluster i , that is,

$$\bar{x}_k(i) = \frac{1}{N(i)} \sum_{j:c(j)=i} x_k(j).$$

We wish to find that assignment of observations to clusters which minimises W (or equivalently, maximises B , the between groups sum of squares).

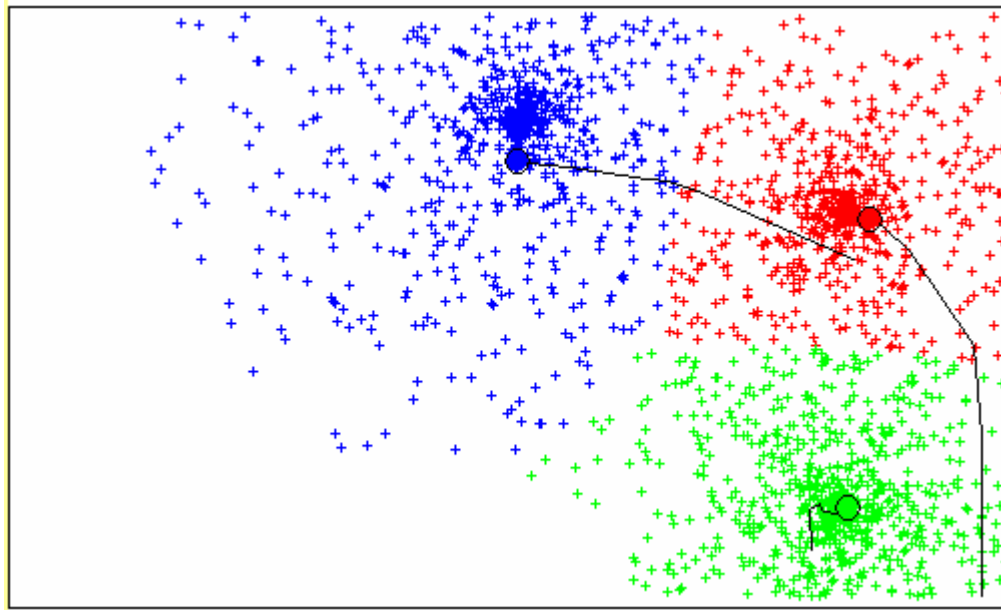
Noting that $\min_m \sum_{j:c(j)=i} \|x(j) - m\|^2$ is achieved by $m = \bar{x}(i)$ (consider partial derivatives w.r.t. the components of m), we have, writing c for the function that assigns observations to clusters,

$$\min_c \sum_{i=1}^k \sum_{j:c(j)=i} \|x(j) - \bar{x}(i)\|^2 = \min_{c; m(1), \dots, m(k)} \sum_{i=1}^k \sum_{j:c(j)=i} \|x(j) - m(i)\|^2.$$

This form of the objective function suggests the following iterative solution technique:

1. Given c put $m(i) = \bar{x}(i)$ for each $i = 1, \dots, k$.
2. Given $m(1), \dots, m(k)$ define $c(j) = i$ where $m(i)$ is the closest to $x(j)$.
3. If c has changed go back to 1.

We start the algorithm at step 2 with $m(1), \dots, m(k)$ chosen at random without replacement from $x(1), \dots, x(N)$.



An example with $k = 3$. Plotted are the paths followed by each centroid.

Each iteration of the algorithm requires kN comparisons to make cluster assignments and $O(N)$ operations to calculate centroids.

It is easy to see that both steps 1 and 2 of the algorithm reduce W . For step 1 we observe as before that $\min_m \sum_{j:c(j)=i} \|x(j) - m\|^2$ is achieved by $m = \bar{x}(i)$. For step 2 note that if $x(j)$ is moved from cluster i to i' , then W increases by

$$\|x(j) - m(i')\|^2 - \|x(j) - m(i)\|^2.$$

It follows that the algorithm must converge, however there is no guarantee that it finds the global minimum. In practice the initial choice of centroids can have a large effect on the outcome of the algorithm, so it is usual to re-run it several times with different starting points. The success of the final outcome is also highly dependent on the initial choice of k .

Note that the clusters form a Voronoi tessellation.

Other methods

In addition to hierarchical and partitioning methods, clustering can be achieved using *density based* methods, such as Gaussian mixtures.