Outline Lecture Notes
Math 5772 2016/17

Chapter 1

Dissimilarity and distance

1.1 Introduction

In this part of the course, we are going to look at two related problems that are more to do with the *distances* between observations than with the observations themselves. In other words, we are interested in comparative measures rather than direct measurements. Particularly, we want to measure the "dissimilarity" between observations — not the same as "distance".

First, given a set of observations, how can we group them into "clusters" when we don't know in advance what clusters might exist? Intuitively, a cluster is a group of observations that are "close" to each other, so we need some idea of "distance" between observations. This is the area of *cluster analysis*.

Secondly, if we have the "dissimilarities" between a set of observations, we would like to be able to display that information graphically. If the dissimilarities are true distances, and we are working in 2 dimensions, then this is easy - draw a picture. We'll see that given a set of Euclidean distances it is always possible to recover the "shape" of the original data. But if instead of distances we have dissimilarities, then how to represent the data in a Euclidean way? To do this, we use a technique called *multi-dimensional scaling.*

Clustering and multi-dimensional scaling both require some measure of the *distance* or, more generally, the *dissimilarity* between two observations $\boldsymbol{x}_r$ and $\boldsymbol{x}_s$. We have that $\boldsymbol{x}_r = (x_{r1}, \ldots, x_{rp})^T$ where x_{ri} is the observed value of variable i for individual r.

How to measure dissimilarity? We use a measure $d : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ with three properties

$$\begin{array}{lrcl}
\textsc{Symmetric} & d(\boldsymbol{x}_r, \boldsymbol{x}_s) & = & d(\boldsymbol{x}_s, \boldsymbol{x}_r) \\
\textsc{Non-negative} & d(\boldsymbol{x}_r, \boldsymbol{x}_s) & \geq & 0 \\
\textsc{zero to self} & d(\boldsymbol{x}_r, \boldsymbol{x}_r) & = & 0
\end{array}$$

If a dissimilarity measure satisfies the *metric inequality*,

$$d(\boldsymbol{x}_r, \boldsymbol{x}_s) + d(\boldsymbol{x}_s, \boldsymbol{x}_t) \geq d(\boldsymbol{x}_r, \boldsymbol{x}_t),$$

then it is called a *distance measure* (i.e. the distance from R to T cannot be made shorter by a detour via S).

So the first step is to take a data matrix $X_{n\times p}$ and create a dissimilarity matrix $D_{n\times n} = (d_{rs})$ where $d_{rs} = d(\boldsymbol{x}_r, \boldsymbol{x}_s)$. MDS and some clustering methods only require D; some clustering methods also require the original data matrix X.

1.2 Continuous data

If all the p variables are continuous, then measuring dissimilarity is relatively straightforward. The main measures are

Euclidean

$$d_E(\boldsymbol{x}_r, \boldsymbol{x}_s) = ||\boldsymbol{x}_r - \boldsymbol{x}_s||_2 = \left[\sum_{i=1}^{p} (x_{ri} - x_{si})^2\right]^{1/2}$$

City-block or Manhattan or Taxi-cab

$$d_C(\boldsymbol{x}_r, \boldsymbol{x}_s) = \sum_{i=1}^{p} |x_{ri} - x_{si}|$$

Pearson distance

$$d_P^2(\boldsymbol{x}_r, \boldsymbol{x}_s) = \sum_{i=1}^{p} \frac{(x_{ri} - x_{si})^2}{s_{ii}} = (\boldsymbol{x}_r - \boldsymbol{x}_s)^T D(\boldsymbol{x}_r - \boldsymbol{x}_s)$$

where $S = (s_{ij})$ is the sample variance matrix of X and $D = \text{diag}(s_{ii}^{-1})$, or Pearson distance is the Euclidean distance between standardised data points.

Mahalanobis distance

$$d_M^2(\boldsymbol{x}_r, \boldsymbol{x}_s) = (\boldsymbol{x}_r - \boldsymbol{x}_s)^T S^{-1}(\boldsymbol{x}_r - \boldsymbol{x}_s),$$

or Mahalanobis distance is the Euclidean distance between decorrelated, standardised data points.

Comments:

1. Formally, d_E and d_C are both special cases of the $\mathbb{L}_q$ norm

$$||\boldsymbol{x}_r - \boldsymbol{x}_s||_q = \left[\sum_{i=1}^{p} |x_{ri} - x_{si}|^q\right]^{1/q}$$

   — Euclidean distance is the $\mathbb{L}_2$ norm and the city-block distance is the $\mathbb{L}_1$ norm.

2. The $\mathbb{L}_q$ norm (and hence both d_E and d_C) satisfies the metric inequality, so these measures are true distance measures.

3. The Mahalanobis measure is very useful in some areas of multivariate analysis, but not so useful in cluster analysis. Why might this be?

4. Choosing a distance measure — basically, use common sense. Think about the following: say we have $\boldsymbol{x}_r = (100, 0), \boldsymbol{x}_s = (110, 0), \boldsymbol{x}_t = (110, 1)$. If, in the context of a particular example, you feel that $d(\boldsymbol{x}_r, \boldsymbol{x}_s)$ should be $\approx d(\boldsymbol{x}_r, \boldsymbol{x}_t)$, then use Euclidean distance. If you feel that $\boldsymbol{x}_t$ is substantially "further" from $\boldsymbol{x}_r$ than $\boldsymbol{x}_s$ is, use city-block. Consider two examples:

   (a) Horticultural data. The two axes are land area given over to wheat and barley. A small change in the amount of barley is not very important compared to the substantial change in wheat – use Euclidean distance.

   (b) Archaeological data. The two axes are number of stone and iron artifacts in a burial site. A change from "no iron item" to "any iron item" is significant – use city-block distance.

1.3 Binary data [extra material]

If the variables are binary, we consider two main options: matching coefficients (symmetric) or the Jaccard coefficient (asymmetric). Say each observation $\boldsymbol{x}_r$ consists of a string of 0's and 1's. For two observations $\boldsymbol{x}_r$ and $\boldsymbol{x}_s$, tabulate the data as a contingency table:

		$\boldsymbol{x}_s$		
		1	0	
$\boldsymbol{x}_r$	1	a	b	$a+b$
	0	c	d	$c+d$
		$a+c$	$b+d$	$a+b+c+d=p$

For example, the observations

$$\begin{aligned} \boldsymbol{x}_r &= \begin{pmatrix}1 & 1 & 0 & 1 & 0 & 0 & 1\end{pmatrix}^T \\ \boldsymbol{x}_s &= \begin{pmatrix}0 & 1 & 1 & 0 & 0 & 1 & 1\end{pmatrix}^T \end{aligned}$$

lead to the contingency table

		$\boldsymbol{x}_s$		
		1	0	
$\boldsymbol{x}_r$	1	2	2	4
	0	2	1	3
		4	3	7

Then the two measures are

$$\text{Matching coefficients} \quad \frac{b+c}{p} = \frac{4}{7} \approx 0.57 \tag{1.3.1}$$

$$\text{Jaccard coefficient} \quad \frac{b+c}{a+b+c} = \frac{4}{6} \approx 0.67. \tag{1.3.2}$$

Another measure that is often used is

$$\text{Sokal and Sneath's coefficient} \ \frac{2(b+c)}{a+2(b+c)} = \frac{8}{10} = 0.8$$

Comments:

1. Basically, the difference between (1.3.1) and (1.3.2) is what we choose to do with matching zeros. Should $x_{ri} = x_{si} = 0$ count for the same as $x_{ri} = x_{si} = 1$?

   YES if this is a binary variable where what we really want to know is “are these two observations the same — e.g. male / female.

   NO if only 1’s hold any real significance — e.g. “is there any family history of haemophilia?” with 1 = yes, 0 = no.

   Sokal and Sneath’s coefficient penalises more for differences. As with continuous data, context is vital.

2. Note that the measures are proportions, and hence will always lie in [0,1]. They also place equal weight on each variable (apart from the fact that a variable that is nearly always zero will rarely contribute to the Jaccard coefficient).

1.4 Categorical data [extra material]

Say variable i takes one of l_i unordered levels (eg blood type; canine subspecies; gene base). We can extend the idea of matching coefficients to this situation. For two individuals r and s, they score $z_{rsi} = 0$ if they “match” on variable i (i.e. if $x_{ri} = x_{si}$) and $z_{rsi} = 1$ otherwise. Then their dissimilarity score is

$$d(\boldsymbol{x}_r, \boldsymbol{x}_s) = \frac{1}{p} \sum_{i=1}^{p} z_{rsi}.$$

NB: Even though one of our examples is genetics this really needs to be modified if we have gene data — see Everitt, Landau & Leese, p.39 — a good example of tweaking our dissimilarity score to allow for the context of the data.

1.5 General issues

There are various complications that can arise. We'll outline some here; for more details see chapter 3 of Everitt, Landau, & Leese.

- Things get a little more complex when we have more than one of continuous, binary, or categorical variables in our data matrix. One suggestions are to dichotomise all the variables and treat them as binary — this leads to loss of a lot of information. Another is to construct a dissimilarity matrix for type of variable and combine them into a single coefficient.

- To standardise or not standardise? If one continuous variable i has a range of 0 - 100 and a second, j, is between 0 and 5, the first will dominate the dissimilarity matrix. We could

  - ignore this and proceed as usual — appropriate if variable i is more important;
  - standardise both by their range (i.e. divide variable i by 100 and variable j by 5 so both are in [0,1]) or by their standard deviation (to get the Pearson distance measure). Now both can contribute equally.

  If we don't standardise when we should, effectively ignoring j. If j is really not important and we standardise, then we are allowing an unimportant variable to colour our judgement.

- More generally, we might believe some variables are more important than others; in that case, we should incorporate weights into our dissimilarity measure.

- Missing values can often occur. Discarding observations with any missing values is appealing but (a) can lead to bias and (b) can result in dropping a lot of data. Multiple imputation can be used — effectively predicting missing values from existing ones — but you have to allow for the added uncertainty.

Chapter 2

Cluster analysis

2.1 Introduction

Clustering attempts to classify observations to groups or *clusters* when the number of groups and group membership of individual observations is *unknown*. This is related to an easier problem — given observations with *known* group membership, how can we distinguish between the groups in order to classify new observations — which is the subject of *discriminant analysis*.

Note that cluster analysis is inherently *exploratory* — we are not going to be in a position to provide definitive answers; we shall not be looking at significance tests for how many clusters there are in a set of data.

Cluster analysis starts with a dissimilarity matrix $D_{n\times n}$ and attempt to divide the data into clusters based on this (some methods also need the original data matrix $X_{n\times p}$). We shall concentrate on *hierarchical* methods, which operate iteratively and construct a nested series of classifications, but other methods exist. Hierarchical methods are *agglomerative* or divisive — we shall concentrate on agglomerative methods, which are more widely used.

Agglomerative methods start off with each observation being its own "cluster". At each step, combine the two most similar clusters to form a new, larger cluster. Continue until all the observations have been combined into one cluster of size n.

Divisive methods start off with one cluster of size n and attempt to find the split into two clusters which are as dissimilar as possible. At each subsequent step, split one existing cluster until eventually there are n clusters of size one.

One key feature of hierarchical methods is that once a decision to merge (in agglomerative techniques) or split (in divisive techniques) a cluster has been made there is no going back — once a mistake has been made, it cannot be unmade.

2.2 Agglomeration

Clearly, we need to extend our ideas about dissimilarity measures from two observations to two clusters. A dissimilarity measure and a rule for determining the dissimilarity of two clusters determines an agglomerative clustering technique.

Assume that we have chosen a dissimilarity measure $d(\boldsymbol{x}_r, \boldsymbol{x}_s)$ and constructed a dissimilarity matrix $D = (d_{rs})$ from a data matrix X, where the rth row of X, $\boldsymbol{x}_r$ is a vector of data on the rth individual. Suppose we have two clusters $A = \{\boldsymbol{x}_{r_1}, \ldots, \boldsymbol{x}_{r_n}\}$ and $B = \{\boldsymbol{x}_{s_1}, \ldots, \boldsymbol{x}_{s_m}\}$, with no observation being in both A and B. Then we can measure the dissimilarity between A and B using one of the following:

Single linkage (aka "nearest neighbour") — The dissimilarity between clusters A and B is the minimum dissimilarity between one element of A and one element of B.

Average linkage — Uses the mean of all $n \times m$ dissimilarities between objects from A and B.

Complete linkage (aka "furthest neighbour") — The dissimilarity between A and B is the maximum dissimilarity between one element of A and one element of B.

We can regard simple and complete linkage as opposite ends of a range of methods. Averages (and other techniques) will generally give answers "somewhere in between" the other two.

2.3 Examples

Consider this set of points in $\mathbb{R}^2$:

Observation	x_1	x_2
1	1	1
2	1	2
3	6	3
4	8	2
5	8	0

The Euclidean distance matrix is

$$D = \begin{bmatrix} 0.00 & & & & \\ 1.00 & 0.00 & & & \\ 5.39 & 5.10 & 0.00 & & \\ 7.07 & 7.00 & 2.24 & 0.00 & \\ 7.07 & 7.28 & 3.61 & 2.00 & 0.00 \end{bmatrix}$$

We can use single linkage agglmerative clustering with Euclidean distance to obtain the following sequence of partitions:

[step 1] Combine 1 & 2 at a dissimilarity of 1. Partition [(1,2), 3, 4, 5].

[step 2] Combine 4 & 5 at a dissimilarity of 2. Partition [(1,2), 3, (4,5)].

[step 3] Combine 3 & (4, 5) at a dissimilarity of 2.24. Partition [(1,2), (3,4,5)].

[step 4] Combine (1,2) & (3,4,5) at a dissimilarity of 5.10. Partition [(1,2,3,4,5)].

There are actually two lots of information here: which elements / clusters are combined at each step and at what dissimilarity the agglomeration occurs. This information is often summarised in a *dendrogram.*

The "leaves" at the bottom correspond to the individuals. Lines going up from the individuals converge when those individuals have been agglomerated into a cluster. The vertical axis shows at what dissimilarity (or "height") the agglomeration occurs.

In fact, all 3 methods give the same sequence of partitions here. (**Exercise** — check this by hand for yourself.) This is not surprising when you look at a plot of the data; elements 1 and 2 are clearly a long way from 3, 4, and 5. However, the dendrograms do look different as the agglomeration occurs at different "heights".

Task: As we've noted, this is a fairly basic data set. See what happens when you change from Euclidean distance to other measures. Also see whether you can get the linkage methods to give different answers by moving one of the data points.

For hints on how to do this in *R*, see the internet and Handout M1.

2.4 *k*-means clustering

Start with an $n \times p$ data matrix X. Suppose that the number k of clusters to be found is known, and suppose that an initial clustering is available, in the form of an initial labelling $\ell^{(0)}(r) \in \{1, \ldots, k\},\ r = 1, \ldots, n.$

The k-means algorithm iteratively refines this labelling in order to find a local minimum of the residual sum of squares

$$RSS = \sum_{j=1}^{k} (\boldsymbol{x}_r - \hat{\boldsymbol{\mu}}_j)^T (\boldsymbol{x}_r - \hat{\boldsymbol{\mu}}_j),$$

where

$$\hat{\boldsymbol{\mu}}_j = \sum_{\substack{r=1 \\ \ell(r)=j}}^{n} \boldsymbol{x}_r \Big/ \sum_{\substack{r=1 \\ \ell(r)=j}}^{n} 1 \tag{2.4.1}$$

denotes the sample mean of the jth cluster using the current labelling.

The k-means algorithm proceeds by the following steps:

1. Given the current labelling, define the sample mean of each cluster.
2. For each $r = 1, \ldots, n$, do the following. Compute the Euclidean distance between $\boldsymbol{x}_r$ and each of the sample means. If $\boldsymbol{x}_r$ is closest to the mean of its current cluster, do not change its label. But if $\boldsymbol{x}_r$ is closer to the sample mean of another cluster, then reset $\ell(r)$ to be the label of that cluster.
3. Repeat steps 1 and 2 until no further change of labels takes place.

Note that the k-means algorithm uses Euclidean distance. Hence some preprocessing of the data may be required, e.g. if the p variables have substantially different variances.

Also, the the final clustering of the algorithm can depend heavily on the initial clustering. In practice it is beneficial to repeat the algorithm from many (e.g. 100) different starting points, and to report the best answer as the final solution.

2.5 Finite mixture density classification

2.5.1 Finite mixture distributions (FMDs)

For example, take a large sample of healthy U.K. adults and measure their heights. A histogram would probably look slightly bimodal (separate peaks for males and females).

Say we wanted to estimate the underlying density $f(h)$ where h = height. How could we do this?

One way would be to use a non-parametric smoothing approach (e.g. kernel density estimation, wavelets). But these methods do not give a functional form for $f(h)$, which we might want.

An alternative is to recognise that there are probably two overlapping normal distributions involved. Hor a new observation of H, we might have

$$H|\text{ individual male } \sim N(173, 100) \qquad\qquad H|\text{ individual female } \sim N(147, 100).$$

This is an example of a *finite mixture distribution.*

Generally, consider a univariate random variable X, which has a FMD if its pdf is of the form

$$f_X(x) = \sum_{j=1}^{c} \pi_j f_j(x)$$

where $\sum_j \pi_j = 1, 0 < \pi_j < 1$ and each individual $f_j(x)$ is a pdf. The π_j are called *mixing weights* and the f_j are the pdf of the *component densities* (and will have their own, different, means, variances, and any other necessary parameters).

In our height example, $f(h)$ is the pdf of a mixture density with two normal components. Let $\phi(h|\mu, \sigma^2)$ be the pdf of a $N(\mu, \sigma^2)$ distribution, then

$$f(h) = \frac{1}{2}\phi(h|147, 100) + \frac{1}{2}\phi(h|173, 100).$$

More generally, if $f_j(\boldsymbol{x}|\boldsymbol{\theta}_j)$ is the pdf of a random vector with parameter vector $\boldsymbol{\theta}_j$, then a c-component mixture model has pdf

$$f(\boldsymbol{x}) = \sum_{j=1}^{c} \pi_j f_j(\boldsymbol{x}|\boldsymbol{\theta}_j).$$

Fitting this density to a sample of data would involve estimating the parameters $\{\pi_j, \boldsymbol{\theta}_j\}$, (they are, the weightings and the component distribution parameters).

2.5.2 FMDs for clustering

We consider the MV c-component normal mixture where $f_j(\boldsymbol{x}|\boldsymbol{\theta}_j) \sim N_p(\boldsymbol{\mu}_j, \Sigma_j)$. In this case, each component represents a cluster (so we have to decide on the number of clusters *before* doing any estimation).

Assume that we have both picked a number of clusters and have estimates of $\pi_j, \boldsymbol{\mu}_j,$, and Σ_j for $j = 1, \ldots, c$. Then the probability that observation $\boldsymbol{x}_r$ is in cluster j is given by Bayes' Theorem:

$$\text{pr}\{\text{cluster } j|\boldsymbol{x}_r\} = \frac{\widehat{\pi}_j f_j(\boldsymbol{x}_r|\widehat{\boldsymbol{\mu}}_j, \widehat{\Sigma}_j)}{\sum_{i=1}^{c} \widehat{\pi}_i f_i(\boldsymbol{x}_r|\widehat{\boldsymbol{\mu}}_i, \widehat{\Sigma}_i)} \tag{2.5.1}$$

We then assign each observation to the cluster with the highest probability.

2.5.3 Estimating parameters

We *outline* the estimation of the necessary parameters given a sample of observations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ from a c-component MV normal FMD. The full details are beyond the scope of this course. Let $\boldsymbol{\theta}$ denote the collection of parameters $\{\pi_j, \boldsymbol{\mu}_j, \Sigma_j\}$, then the log-likelihood of $\boldsymbol{\theta}$ is

$$\begin{aligned} L(\boldsymbol{\theta}; X) &= \sum_{r=1}^{n} \log f(\boldsymbol{x}_r|\boldsymbol{\theta}) + c \\ &= \sum_{r=1}^{n} \log \left\{ \sum_{j=1}^{c} \frac{\pi_j}{\sqrt{(2\pi)^p|\Sigma|}} \exp\left[-\frac{1}{2}(\boldsymbol{x}_r - \boldsymbol{\mu}_j)^T \Sigma_j^{-1} (\boldsymbol{x}_r - \boldsymbol{\mu}_j) \right] \right\} + c. \end{aligned}$$

Given probabilities of an observation $\boldsymbol{x}_r$ being from cluster j from 2.5.1, the maximum likelihood estimates are

$$\widehat{\pi}_j = \frac{1}{n} \sum_{r=1}^{n} \mathrm{pr}\{\text{cluster } j|\boldsymbol{x}_r\} \qquad (2.5.2)$$

$$\widehat{\boldsymbol{\mu}}_j = \frac{1}{n\widehat{\pi}_j} \sum_{r=1}^{n} \boldsymbol{x}_r \mathrm{pr}\{\text{cluster } j|\boldsymbol{x}_r\} \qquad (2.5.3)$$

$$\widehat{\Sigma}_j = \frac{1}{n} \sum_{r=1}^{n} (\boldsymbol{x}_r - \boldsymbol{\mu}_j)(\boldsymbol{x}_r - \boldsymbol{\mu}_j)^T \mathrm{pr}\{\text{cluster } j|\boldsymbol{x}_r\}. \qquad (2.5.4)$$

Given initial estimates of $\boldsymbol{\theta}$, we can evaluate the probabilities in (2.5.1) and use these to re-estimate $\boldsymbol{\theta}$ by evaluating (2.5.2)–(2.5.4) for $j = 1, \ldots, c$. the process is then iterated until the estimates converge. For this to work well, the Σ_j are usually assumed to be equal.

Notes:

1. Chapter 6 of Everitt *et al.*'s "Cluster Analysis" gives more on deriving the MLEs.
2. This algorithm is an example of something called an "EM" algorithm; given Expected values (from (2.5.1)), we perform a Maximisation to get the MLEs in (2.5.2)-(2.5.4).
3. Approximate likelihood ratio tests for the number of clusters are available but are fairly complicated (and very approximate). We use a modified L.R. test statistic to test H_0:c components against H_1:$c+1$ components. If λ is the usual L.R. test statistic, then the modified version is

   $$\lambda^* = -\frac{2\log\lambda}{n}\left(n - 1 - p - \frac{c+1}{2}\right).$$

   It is believed that under H_0, $\lambda^* \sim \chi^2_{2q+1}$ (this has not (yet) been proven, but simulations imply that this seems plausible).

4. An alternative way of selecting a model is to maximise the *Bayes Information Criterion* or BIC, given by

   $$\mathrm{BIC} = 2L(\boldsymbol{\theta}|X) - k\log n$$

   where k is the number of parameters to be estimated and n the number of observations. The function `Mclust` in the `mclust` *R* library fits a normal mixture cluster model using the BIC to select the number and type of clusters.

5. The components can, in principle, be from different distributions.

Chapter 3

Multi-dimensional scaling (MDS)

3.1 Introduction

Start with an $n \times n$ *dissimilarity matrix* $\Delta = (\delta_{rs})$ representing the pairwise dissimilarities between a collection of n objects. That is Δ is a symmetric matrix with nonnegative entries and a vanishing diagonal,

$$\delta_{rr} = 0, \quad \delta_{rs} = \delta_{sr} \geq 0,\ r, s = 1, \ldots, n.$$

To understand the structure in Δ, it is helpful to display the information in Δ graphically. That is, we want to find a set of points in a low-dimensional Euclidean space, $\{\boldsymbol{x}_r \in \mathbb{R}^k,\ r = 1, \ldots, n\}$, (typically $k = 1, 2$ or occasionally 3) such that the Euclidean interpoint distance matrix

$$D = (d_{rs}), \quad d_{rs}^2 = (\boldsymbol{x}_r - \boldsymbol{x}_s)^T (\boldsymbol{x}_r - \boldsymbol{x}_s)$$

for X matches Δ as closely as possible. The method used to tackle this problem is called *classic multi-dimensional scaling* or MDS:

[There is also a more nonparametric version called non-metric MDS, or ordinal MDS, which only preserves the *ordering* of the δ_{rs} and otherwise ignores their magnitudes. Non-metric MDS will not be studied here.]

A special case is when we attempt a 1D solution — here, we are essentially trying to put all the individuals in some sort of order, This problem is known as *seriation* and is a common goal in archaeological statistics.

3.2 Mathmatical motivation for MDS

This section gives the mathematical motivation behind classic approach to multidimensional scaling.

3.2.1 Centered inner product lemma

Let $X(n \times p)$ be a data matrixk, i.e. a configuration of n points in $\mathbb{R}^p$. Define the following matrices from X. Remember $H = I_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T$ is the $n \times n$ *centering matrix.* It is symmetric ($H^T = H$) and idempotent ($H^2 = H$).

1. *Centered data matrix for X.* Let $Y = HX$ denote the centered version of X. The p-dimensional rows $\boldsymbol{y}_r^T$ of Y, when written as column vectors, are given by $\boldsymbol{y}_r = \boldsymbol{x}_r - \overline{\boldsymbol{x}},\ r = 1, \ldots, n.$

2. *Centered inner product matrix for X.* The *centered inner product matrix* $C(n \times n)$, say, is defined by the inner products between the rows of Y,

   $$c_{rs} = \boldsymbol{y}_r^T \boldsymbol{y}_s.$$

   In matrix notation

   $$C = YY^T.$$

   Caution. Notice the difference between the $n \times n$ matrix $C = YY^T$, the inner product matrix between the *rows* of Y, and the $p \times p$ sample covariance matrix $S = Y^T Y/(n-1) = X^T HX/(n-1)$, which is proportional to the inner product matrix between the *columns* of Y. In particular, with $n > p$, the sample covariance matrix is generally positive definite. On the other hand, C must always have at least one zero eigenvalue, with eigenvector $\mathbf{1}_n$ ($C\mathbf{1}_n = \mathbf{0}_n$ since $H\mathbf{1}_n = \mathbf{0}_n$).

3. *Euclidean distance matrix for X.* The squared Euclidean distances $d_{rs}^2,\ r = 1, \ldots, n$ between the rows of X are defined by

   $$\begin{aligned} d_{rs}^2 &= (\boldsymbol{x}_r - \boldsymbol{x}_s)^T(\boldsymbol{x}_r - \boldsymbol{x}_s) \\ &= (\boldsymbol{y}_r - \boldsymbol{y}_s)^T(\boldsymbol{y}_r - \boldsymbol{y}_s) \\ &= c_{rr} + c_{ss} - 2c_{rs}. \end{aligned}$$

   For matrix calculations it is helpful to define a matrix

   $$A = (a_{rs}), \quad a_{rs} = -\frac{1}{2}d_{rs}^2$$

   containing minus 1/2 times the squared Euclidean distances.

4. *Doubly centered rescaled squared Euclidean distance matrix.* Let

$$B = HAH$$

denote the double-centered version of A.

5. *Doubly centered rescaled squared dissimilarity matrix.* Let Δ denote an $n \times n$ dissimilarity matrix. That is, the diagonal elements vanish, $\delta_{rr} = 0,\ r = 1, \ldots, n$ and the off-diagonal elements are nonnegative and symmetric, $\delta_{rs} = \delta_{sr} \geq 0$. From Δ it is useful to construct a *double centered rescaled squared dissimilarity matrix* $B = B(\Delta)$ as follows. First define an $n \times n$ matrix A with entries

$$a_{rs} = -\frac{1}{2}\delta_{rs}^2.$$

Then define

$$B = HAH$$

to be the double-centered version of A.

Lemma 1 (Centered inner product lemma). Let X be an $n \times p$ configuration. With the above notation, the centered innner product matrix and the doubly centered rescaled squared Euclidean distance matrix are identical,

$$B = C.$$

Proof. Define an n- vector $\boldsymbol{v}$ by

$$v_r = \boldsymbol{y}_r^T \boldsymbol{y}_r,$$

so that $\boldsymbol{v}$ contains the inner products of the rows of Y with themselves. Then

$$-\frac{1}{2}d_{rs}^2 = -\frac{1}{2}\{(\boldsymbol{y}_r - \boldsymbol{y}_s)^T(\boldsymbol{y}_r - \boldsymbol{y}_s)\} = v_r + v_s - 2c_{rs},$$

or in matrix form

$$A = -\frac{1}{2}\{\boldsymbol{v}\mathbf{1}_n^T + \mathbf{1}_n\boldsymbol{v}^T - 2C\}.$$

Pre- and post-multiplying by H and remembering $H\mathbf{1}_n = \mathbf{0}_n$ yields $B = C$, as required.

3.2.2 Inverse centering lemma

In this section we show that double-centering a dissimilarity matrix does not lose any information (because the diagonal entries vanish).

Lemma 2 (Inverse centering lemma). Let $A(n \times n)$ be a symmetric matrix with a vanishing diagonal, $a_{rr} = 0,\ r = 1, \ldots, n$. Set $B = HAH$ to be the doubly-centered version of A. Then the elements of A can be recovered from B.

Proof and construction. We shall use the notation

$$a_{r\cdot} = \frac{1}{n}\sum_{s=1}^{n} a_{rs}, \quad a_{\cot s} = \frac{1}{n}\sum_{r=1}^{n} a_{rs}, \quad a_{\cdot\cdot}\frac{1}{n^2}\sum_{r,s=1}^{n} a_{rs}$$

to be the sums over the rows and columns, and both, of A. Let $\boldsymbol{u} = (u_r)$ be a vector with elements $u_r = a_{r\cdot} = a_{\cdot r}$. Note $\frac{1}{n}\sum u_r = a_{\cdot\cdot}$.

Then

$$B = (I - \frac{1}{n}\mathbf{1}\mathbf{1}^T)A((I - \frac{1}{n}\mathbf{1}\mathbf{1}^T) == A - \boldsymbol{u}\mathbf{1}^T - \mathbf{1}\boldsymbol{u}^T + a_{\cdot\cdot}\mathbf{1}\mathbf{1}^T,$$

or in terms of the elements of B,

$$b_{rs} = a_{rs} - u_r - u_s + a_{\cdot\cdot} = a_{rs} - a_{r\cdot} - a_{\cdot s} + a_{\cdot\cdot}. \tag{3.2.1}$$

Then we proceed in 3 steps to recover A from B.

(a) Sum the diagonal terms in (3.2.1) to get

$$\frac{1}{n}b_{rr} = 0 - a_{\cdot\cdot} - a_{\cdot\cdot} + a_{\cdot\cdot} = a_{\cdot\cdot}$$

since $a_{rs} = 0$ for $r = s$. Hence $a_{\cdot\cdot}$ can be found from B.

(b) Next set $r = s$ in (3.2.1) to get

$$b_{rr} = 0 - 2a_{r\cdot} + a_{\cdot\cdot}.$$

Hence a_{rr} can be found from B and the fact that we have already found $a_{\cdot\cdot}$.

(c) Finally, for a general index pair r, s, (3.2.1) can be used to find a_{rs} from B.

Corollary 1. If two matrices $A^{(1)}$ and $A^{(2)}$ generate matrices $B^{(1)} = HA^{(1)}H$ and $B^{(2)} = HA^{(2)}H$, and if $B^{(1)} = B^{(2)}$, then $A^{(1)} = A^{(2)}$.

Corollary 2. If two dissimilarity matrices $\Delta^{(1)}$ and $\Delta^{(2)}$ generate the same B-matrices ($B^{(i)} = HA^{(i)}H$, where $A^{(i)} = (a^{(i)}_{rs})$ with $a^{(i)}_{rs} = \frac{1}{2}\delta^{(i)2}_{rs}$, $i = 1, 2$, then $\Delta^{(1)} = \Delta^{(2)}$.

3.2.3 Main Torgerson Gower Theorem

Start with a dissimilarity matrix Δ and define matrices A and B by $a_{rs} = \frac{1}{2}\delta^2_{rs}$ and $B = HAH$.

A dissimilarity matrix Δ is called *Euclidean* (or a Euclidean distance matrix) if there exists an $n \times q$ configuration X for some dimension $q \geq 1$ such that the elements of Δ are the Euclidean distances between the rows of X,

$$\delta^2_{rs} = (\boldsymbol{x}_r - \boldsymbol{x}_s)^T(\boldsymbol{x}_r - \boldsymbol{x}_s), \quad r, s = 1, \ldots, n.$$

Theorem 1. (Torgerson Gower theorem). A dissimilarity matrix Δ is Euclidean if and only if the doubly centered rescaled squared dissimilarity matrix B is positive semi-definite (p.s.d.)

Proof.

If Δ is Euclidean, there exists an $n \times q$ configuration X for some $q \geq 1$ such that the Euclidean distances between the rows of X are given by Δ. Hence, B is minus 1/2 times the doubly centered rescaled squared Euclidean distance matrix for X. By the centered inner product lemma, $B = C = YY^T$, where $Y = HX$ is the centered configuration.

We want to show that B is p.s.d. Let $\boldsymbol{a}$ be an n-vector of coefficients. Then

$$\boldsymbol{a}^T B \boldsymbol{a} = \boldsymbol{a} Y Y^T \boldsymbol{a} = ||Y^T \boldsymbol{a}||^2 \geq 0,$$

This inequality holds for all $\boldsymbol{a}$; hence B is p.s.d.

Conversely, suppose B is p.s.d. and consider its spectral decomposition $B = \Gamma \Lambda \Gamma^T$ where $\Gamma(n \times n)$ is orthogonal and $\Lambda(n \times n)$ is diagonal with nonnegative entries. Suppose $1 \leq q \leq n$ of the eigenvalues are positive and partition

$$\Lambda = \begin{bmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_2 \end{bmatrix}, \quad \Gamma = \begin{bmatrix} \Gamma_1 & \Gamma_2 \end{bmatrix}$$

where $\Lambda_1(k \times k)$ contains the nonzero eigenvalues and $\Lambda_2 = 0$ of size $(n-k) \times (n-k)$ contains the zero eigenvalues. Similarly partition Γ where $\Gamma_1(n \times k)$ contains the eigenvectors for Λ_1.

Then the the spectral decomposition can be written in "reduced" form as

$$B = \Gamma_1 \Lambda_1 \Gamma_1^T$$

(check!). Set

$$Z = \Gamma_1 \Lambda_1^{1/2}$$

where $\Lambda_1^{1/2} = \text{diag}(\lambda_j^{1/2})$ contains the square roots of the positive eigenvalues.

Then Z of size $n \times q$ is the desired configuration. To confirm this result, first note that

$$B = \Gamma_1 \Lambda_1 \Gamma_1^T = \Gamma_1 \Lambda_1^{1/2} \Lambda_1^{1/2} \Gamma_1^T = ZZ^T$$

is the inner product matrix for Z.

Second, note that $\mathbf{1}_n$ is an eigenvector of B with eigenvalue 0 (since $H\mathbf{1}_n = \mathbf{0}_n$ implies $B\mathbf{1}_n = \mathbf{0}_n$). Since the eigenvalues of a symmetric matrix for distinct eigenvalues are orthogonal to on another, the columns of Z must be orthognal to $\mathbf{1}_n$, i.e. $\boldsymbol{z}_{(j)}^T \mathbf{1}_n = 0,\ j = 1, \ldots, q$. Hence $Z = HZ$, i.e. Z is a centered configuration.

Thus B is also the *centered inner product matrix* for Z. Hence by the centered inner product lemma, B is also minus 1/2 times the doubly centered squared Euclidean distance matrix for Z. Finally, Corollary 2 for the inverse centering theorem tells us that Δ is the Euclidean distance matrix for Z.

3.3 Classical MDS

In applied statistical work, the data take the form an $n \times n$ dissimilarity matrix Δ. Generally, the double centered rescaled squared dissimilarity matrix $B = B(\Delta)$ is not p.s.d. However, we "hope" that for some small value of k (typically $k = 1, 2$ or 3) that

- the first k eigenvalues will be "large" and positive, and
- the remaining eigenvalues will have "small" absolute values.

That is, Δ can be thought of as a "noisy" version of a rank k p.s.d. matrix. Let $B = \Gamma\Lambda\Gamma^T$ denote the spectral decomposition of B. Partition

$$\Lambda = \begin{bmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_2 \end{bmatrix}, \quad \Gamma = \begin{bmatrix} \Gamma_1 & \Gamma_2 \end{bmatrix}$$

where $\Lambda_1 (k \times k)$ contains the "large" eigenvalues and Λ_2 of size $(n-k) \times (n-k)$ contains the remaining eigenvalues. Similarly partition Γ where $\Gamma_1 (n \times k)$ contains the eigenvectors for Λ_1. Define a new configuration

$$X = \Gamma_1 \Lambda_1^{1/2}$$

Then we "expect" the Euclidean distances between the rows of Z to be approximately equal to the elements of Δ. [By the Torgerson Gower Theorem this property would be exact if the eigenvalues in Λ_2 were all exactly equal to 0.]

Finally, we plot the configuration X as n points in $\mathbb{R}^k$ to get a visual interpretation of the data.

3.4 Connection between MDS & PCA

Imagine we were given a data matrix $X_{n\times p}$ with centered version $Y = HX$ and with associated Euclidean distance matrix $D = (d_{rs})$. Suppose we wished to do classical MDS on X (we might want a lower-dimensional representation of X). Then we would calculate $B = YY^T$ and look at the spectral decomposition of B, i.e. of YY^T. An alternative approach is principal component analysis (PCA), where we look at the spectral decomposition of the scaled sample variance, which is proportional to Y^TY. We now look at the connection between these methods.

Let $\{\lambda_i, \boldsymbol{\gamma}_{(i)}\}$ be the ordered eigenvalues and corresponding unit eigenvectors of Y^TY. Then for $i = 1, \ldots, p$,

$$(Y^TY)\boldsymbol{\gamma}_{(i)} = \lambda_i \boldsymbol{\gamma}_{(i)} \implies (YY^T)Y\boldsymbol{\gamma}_{(i)} = \lambda_i Y\boldsymbol{\gamma}_{(i)},$$

so λ_i must also be an eigenvalue of YY^T, with corresponding unit eigenvector $\boldsymbol{e}_i \propto Y\boldsymbol{\gamma}_{(i)}$. Moreover, the other $n-p$ eigenvalues of YY^T are zero because Y is of rank p.

Thus, the results of PCA (starting from a data matrix) can be matched to those of classical MDS (starting from the Euclidean distances calculated from that data matrix). But note that if the data consist just of a distance matrix D rather than a configuration X, then PCA is not available.

3.5 Seriation and the horseshoe effect

One major use of MDS is to try to put a collection of objects in order — i.e. a 1D solution. For example, archaeological finds, subspecies of animal, or literary works might be ordered. This is called *seriation* or *unidimensional scaling*.

Imagine we have some dissimilarity measure between the works of Shakespeare and we have used CMDS to produce a 1d representation. The ordering has a "direction" if we know the first and last works. This is often not the case — the method can give us an order, but the direction needs context.

We will consider a simple example from Mardia, Kent & Bibby, who present data on the presence/absence (1/0) of $p = 5$ types of pottery in $n = 6$ burial sites (labelled A–F, say). The data matrix is

$$X = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Just counting the number of matches gives us the similarity matrix

$$M = \begin{bmatrix} 5 & 0 & 3 & 5 & 1 & 3 \\ & 5 & 2 & 0 & 4 & 2 \\ & & 5 & 3 & 1 & 3 \\ & & & 5 & 1 & 3 \\ & & & & 5 & 3 \\ & & & & & 5 \end{bmatrix}.$$

To carry out MDS, we need to transform M to a dissimilarity matrix. Since we like to use Euclidean distance matrices whenever possible, the following theorem is helpful.

theorem[The "Standard Transformation"] The matrix $M = (m_{rs})$ is p.s.d. if and only if the dissimilarity matrix $\Delta = (\delta_{rs})$ given by $d\delta_{rs}^2 = m_{rr} - 2m_{rs} + m_{ss}$ is Euclidean.

Proof: Essentially the same as the Torgerson-Gower Theorem.

Note that for presence/absence data, $m_{rr} = p$ as an object will match itself on each variable. Hence $\max(d_{rs}) = \sqrt{2p}$, which occurs when $m_{rs} = 0$ — when objects r and s have nothing in common. Conversely, if objects r and s have everything in common, then $m_{rs} = p$ and $d_{rs} = 0$. This is intuitively appealing. For our example, the d_{rs} values will be between 0 and $\sqrt{10}$.

Applying the standard transformation to our example gives

$$D = \begin{bmatrix} 0 & \sqrt{10} & 2 & 0 & \sqrt{8} & 2 \\ & 0 & \sqrt{6} & \sqrt{10} & \sqrt{2} & \sqrt{6} \\ & & 0 & 2 & \sqrt{8} & 2 \\ & & & 0 & \sqrt{8} & 2 \\ & & & & 0 & 2 \\ & & & & & 0 \end{bmatrix}$$

As an aside, note that clustering on these data gives the following agglomerations. For single linkage,

1. (A,D), B, C, E, F
2. (A,D), (B,E), C, F
3. (A,B,C,D,E,F) — all combine at once due to ties.

Complete linkage is also affected by the ties:

1. (A,D), B, C, E, F

2. (A, D), (B, E), C, F
3. (A, C, D, F), (B, E)
4. (A, B, C, D, E, F)

Doing MDS on D gives eigenvalues $\lambda_1 = 1.75, \lambda_2 = 0.59, \lambda_3 = 0.35, \lambda_4 = 0.05$, $\lambda_5 = \lambda_6 = 0$. So a 4-dimensional solution is a perfect fit, but we can consider a 1- or 2-dimensional approximate solution. The first two dimensions are

	A	B	C	D	E	F
First axis	-0.60	0.77	-0.19	-0.60	0.64	-0.01
Second axis	-0.15	0.20	0.60	-0.15	-0.35	-0.14,

so we get a seriation of (A,D), C, F, E, B. However, the second dimension values are substantially non-zero in relation to the first axis values (the range on the first axis is 1.37, as compared to 0.95 on the second). This puts a 1D solution in doubt:

Sometimes a 2D solution can be a 1D solution in disguise. This is called the *horseshoe effect*. In our example, note that is is fairly easy for two objects to have a similarity of zero and hence a maximum distance of $\sqrt{2p}$. So in a large sample of objects, we can get a lot of "maximum distance" pairs, all tying. This means that once you hit a "threshold" can't get any more distant. This can result in our not being able to distinguish between "moderate" and "large" distances.

The classic example of the horseshoe effect is due to Kendall (1971). Start with a similarity matrix $M_{51\times 51}$ defined by $m_{rr} = 9$, and

$$m_{rs} = \begin{cases} 8 & 1 \leq |r-s| \leq 3 \\ 7 & 4 \leq |r-s| \leq 6 \\ \vdots & \vdots \\ 1 & 22 \leq |r-s| \leq 24 \\ 0 & |r-s| \geq 25. \end{cases}$$

Here, there is a very clear seriation. Any object can be well-ordered with respect to the objects "near" to it in the ordering. However, objects "further away" do not appear to be as far as they actually are, causing the plot to wrap round.

Chapter 4

Compositional data

4.1 Introduction

Compositional data are a type of multivariate data where all the values are nonnegative and sum to 1.

Example Five rocks have been analysed to check their metal composition in terms of iron, nickel and other metals. The proportions of each metal are given by the rows of the following data matrix.

$$X = \begin{pmatrix} 0.25 & 0.12 & 0.63 \\ 0.21 & 0.11 & 0.68 \\ 0.29 & 0.12 & 0.59 \\ 0.19 & 0.10 & 0.71 \\ 0.29 & 0.14 & 0.57 \end{pmatrix}$$

Note that each row sums to 1.

We can calculate the correlation matrix:

$$R = \begin{pmatrix} 1.00 & 0.87 & -0.99 \\ & 1.00 & -0.93 \\ & & 1.00 \end{pmatrix}$$

□

These data cause problems for standard statistical methods because:

1. The data are bounded unlike the most-used MV distribution.

2. There is at least one perfect linear relationship in the variables.

3. Each row (or coumn) of the correlation matrix must necessarily include at least one negative correlation, which complicates interpretation.

4.2 Ternary plots

In fact, p-dimensional compositional data are defined on a special subset of $\mathbb{R}^p$: the simplex

$$\mathcal{S}^p = \left\{ \mathbf{x} = [x_1, x_2, \ldots, x_p] \in \mathbb{R}^p \middle| x_i > 0, i = 1, 2, \ldots, p; \sum_{i=1}^{p} x_i = \kappa \right\}.$$

Effectively, the data points are in $p-1$ dimensions because if we know the value for $p-1$ of the variables, we can calculate the value of the other using:

$$x_i = \kappa - \sum_{j \neq i} x_j.$$

A common way of displaying compositional data is through a ternary plot that takes advantage of the data being constrained on the simplex. For $p = 3$, we just need a single triangle.

4.3 Transformations

There are distributions that have simplex support (e.g., the Dirichlet distribution), but, by using them, we lose the great number of techniques that have been devised for unbounded or MVN datasets. As already stated, the data are effectively in $p-1$ dimensions so a transformation would be useful. However, a simple linear transformation would not remove the problems with bounds or forced correlation. One way to try to avoid these is to employ a non-linear transformation.

There are many different transformation that can be used, and we will focus on the additive log-ratio:

$$\begin{aligned} \mathbf{y} &= (y_1, \ldots, y_{p-1})^T = \text{alr}(\mathbf{x}) \\ &= \left[\log\left(\frac{x_1}{x_p}\right), \ldots, \log\left(\frac{x_{p-1}}{x_p}\right) \right]^T \end{aligned}$$

(here, $\mathbf{y} \in \mathbb{R}^{\,p-1}$ and note that the ordering of the variables is arbitrary). This transformation tends to be useful because it directly utilises the fact that compositional data gives us information about relative size alone. We may then proceed to use methods and distributions that are defined for unbounded real spaces.

There is one final problem with compositional data that we have not considered and that is when one of the observed variables is zero the transformation cannot be applied. This is often avoided by changing the zeros to ϵ or ignoring the data points altogether.