Chapter 4: Multidimensional Scaling

Contents

1. Introduction

Multidimensional scaling is a method to produce a low-dimensional “map” of the data similar to PCA, but do not operate directly on the usual multivariate data matrix, $\boldsymbol{X}$. Instead they are applied to distance matrices, which are derived from the matrix $\boldsymbol{X}$, and also to so-called “dissimilarity matrices” or “similarity matrices” that arise directly in a number of ways, in particular from judgments made by human raters about how alike pairs of objects, stimuli, etc., of interest are. The term “proximity” is often used to encompass both dissimilarity and similarity ratings.

Multidimensional scaling is essentially a **data reduction technique** because the aim is to find a set of points in low dimension that approximate the possibly high-dimensional configuration represented by the original proximity matrix.

Models for proximity data

Models are fitted to proximities in order to clarify, display, and help understand and possibly explain any structure or pattern amongst the observed or calculated proximities not readily apparent in the collection of numerical values.

In some areas, particularly psychology, the ultimate goal in the analysis of a set of proximities is more specific, namely the development of theories for explaining similarity judgments; in other words, trying to answer the question, “what makes things seem alike or seem different?" Models for the analysis of proximity data can be categorized into one of three major classes: **spatial models**, **tree models**, and **hybrid models**. In this chapter we only deal with the first of these three classes.

Spatial models for proximities: Multidimensional scaling (MDS)

A spatial representation of a proximity matrix consists of a set of n m-dimensional coordinates, each one of which represents one of the n units in the data. The required coordinates are generally found by minimizing some measure of “fit” between the distances implied by the coordinates and the observed proximities. In simple terms, a geometrical model is sought in which the larger the observed distance or dissimilarity between two units (or the smaller their similarity), the further apart should be the points representing them in the model. In general (but not exclusively), the distances between the points in the spatial model are assumed to be Euclidean. Finding the best-fitting set of coordinates and the appropriate value of m needed to adequately represent the observed proximities is the aim of the many methods of multidimensional scaling that have been proposed. The hope is that the number of dimensions, m, will be small, ideally two or three, so that the derived spatial configuration can be easily plotted. The variety of methods that have been proposed largely differ in how agreement between fitted distances and observed proximities is assessed. In this chapter, we will consider two methods, “classical multidimensional scaling” and “non-metric multidimensional scaling”.

2. Classical MDS

Classical scaling seeks to represent a proximity matrix by a simple geometrical model or map. Such a model is characterized by a set of points $\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n$, in m dimensions, each point representing one of the units of interest, and a measure of the distance between pairs of points. The objective of MDS is to determine both the dimensionality, m, of the model, and the n m-dimensional coordinates, $\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n$, so that the model gives a "good" fit for the observed proximities. Fit will often be judged by some numerical index that measures how well the proximities and the distances in the geometrical model match. In essence, this simply means that the larger an observed dissimilarity between two stimuli (or the smaller their similarity), the further apart should be the points representing them in the final geometrical model.

The question now arises as to **how we estimate $\boldsymbol{m}$, and the coordinate values $\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n$, from the observed proximity matrix.** Classical scaling provides an answer to this question. To begin, we must note that there is no unique set of coordinate values that give rise to a set of distances since the distances are unchanged by shifting the whole configuration of points from one place to another or by rotation or reflection of the configuration. In other words, we cannot uniquely determine either the location or the orientation of the configuration. The location problem is usually overcome by placing the mean vector of the configuration at the origin. The orientation problem means that any configuration derived can be subjected to an arbitrary orthogonal transformation. Such transformations can often be used to facilitate the interpretation of solutions, as will be seen later.

Technical details

Assume that the proximity matrix, we are dealing with, is a matrix of Euclidean distances, $\boldsymbol{D}$, derived from a raw $n \times q$ data matrix, $\boldsymbol{X}$. In Chapter 1, we saw how to calculate Euclidean distances from $\boldsymbol{X}$; classical multidimensional scaling is essentially concerned with the reverse problem: given the distances, how do we find X?

First assume $\boldsymbol{X}$ is known and consider the $n \times n$ inner products matrix, $\boldsymbol{B}$

$$\boldsymbol{B} = \boldsymbol{X}\boldsymbol{X}' \qquad (4.1)$$

The elements of $\boldsymbol{B}$ are given by

$$b_{ij} = \sum_{k=1}^{q} x_{ik} x_{jk} \qquad (4.2)$$

It is easy to see that the squared Euclidean distances between the rows of $\boldsymbol{X}$ can be written in terms of the elements of $\boldsymbol{B}$ as

$$d_{ij}^2 = b_{ii} + b_{jj} - 2b_{ij} \qquad (4.3)$$

If the bs could be found in terms of the ds in the equation (4.3), then the required coordinate values could be derived by factoring B as in (4.1). No unique solution exists unless a location constraint is introduced; usually the centre of the points $\bar{\boldsymbol{x}}$ is set at the origin, so that $\sum_{i=1}^{n} x_{ik} = 0$ for all

$k = 1, \dots, m$. These constraints and the relationship given in (4.2) imply that the sum of the terms in any row of $\boldsymbol{B}$ must be zero. Consequently, summing the relationship given in (4.2) over i, over j, and finally over both i and j leads to the series of equations

$$\sum_{i=1}^{n} d_{ij}^2 = T + nb_{jj}, \quad \sum_{j=1}^{n} d_{ij}^2 = T + nb_{ii}, \quad \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}^2 = 2nT$$

where $T = \sum_{i=1}^{n} b_{ii}$ is the trace of the matrix $\boldsymbol{B}$. The elements of $\boldsymbol{B}$ can now be found in terms of squared Euclidean distances as

$$b_{ij} = -\frac{1}{2}\left(d_{ij}^2 - d_{i.}^2 - d_{.j}^2 + d_{..}^2\right)$$

where

$$d_{i.}^2 = \frac{1}{n}\sum_{j=1}^{n} d_{ij}^2, \quad d_{.j}^2 = \frac{1}{n}\sum_{i=1}^{n} d_{ij}^2, \quad d_{..}^2 = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}^2$$

Having now derived the elements of $\boldsymbol{B}$ in terms of Euclidean distances, it remains to factor it to give the coordinate values. In terms of its spectral decomposition (see Chapter 3), $\boldsymbol{B}$ can be written as

$$\boldsymbol{B} = \boldsymbol{V\Lambda V}'$$

where $\boldsymbol{\Lambda} = \text{diag}\,(\lambda_1, \dots, \lambda_n)$ is the diagonal matrix of eigenvalues of $\boldsymbol{B}$ and $\boldsymbol{V} = (\boldsymbol{V}_1, \dots, \boldsymbol{V}_n)$ the corresponding matrix of eigenvectors, normalized so that the sum of squares of their elements is unity, that is, $\boldsymbol{V}_i\boldsymbol{V}_i' = 1$. The eigenvalues are assumed to be labeled such that $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$. When $\boldsymbol{D}$ arises from an $n \times q$ matrix of full rank, then the rank of $\boldsymbol{B}$ is q, so that the last $n - q$ of its eigenvalues will be zero. So $\boldsymbol{B}$ can be written as

$$\boldsymbol{B} = \boldsymbol{V}_1\boldsymbol{\Lambda}_1\boldsymbol{V}_1'$$

where $\boldsymbol{V}_1$ contains the first q eigenvectors and $\boldsymbol{\Lambda}_1$ the q non-zero eigenvalues.
The required coordinate values are thus

$$\boldsymbol{X} = \boldsymbol{V}_1\boldsymbol{\Lambda}_1^{\frac{1}{2}}$$

where $\boldsymbol{\Lambda}^{\frac{1}{2}} = \text{diag}\left(\lambda_1^{\frac{1}{2}}, \dots, \lambda_{\text{q}}^{\frac{1}{2}}\right)$.

Using all q-dimensions will lead to complete recovery of the original Euclidean distance matrix. The best-fitting m-dimensional representation is given by the m eigenvectors of $\boldsymbol{B}$ corresponding to the m largest eigenvalues. The adequacy of the m-dimensional representation can be judged by the size of the criterion

$$P_m = \frac{\sum_{i=1}^{m} \lambda_i}{\sum_{i=1}^{n} \lambda_i}$$

Values of P_m of the order of 0.8 suggest a reasonable fit.

It should be mentioned here that where the proximity matrix contains Euclidean distances calculated from an $n \times q$ data matrix $\boldsymbol{X}$, classical scaling can be shown to be equivalent to principal components analysis, with the required coordinate values corresponding to the scores on the principal component extracted from the covariance matrix of the data. One result of this duality is that classical multidimensional scaling is also referred to as principal coordinates. And the m-dimensional principal components solution ($m < q$) is "best" in the sense that it minimizes the measure of fit

$$S = \sum_{i=1}^{n} \sum_{j=1}^{n} \left(d_{ij}^2 - \left(d_{ij}^{(m)} \right)^2 \right)$$

where d_{ij} is the Euclidean distance between individuals i and j based on their original q variable values and $d_{ij}^{(m)}$ is the corresponding distance calculated from the m principal component scores. When the observed proximity matrix is not Euclidean, the matrix $\boldsymbol{B}$ is not positive-definite. In such cases, some of the eigenvalues of $\boldsymbol{B}$ will be negative; correspondingly, some coordinate values will be complex numbers. If, however, $\boldsymbol{B}$ has only a small number of small negative eigenvalues, a useful representation of the proximity matrix may still be possible using the eigenvectors associated with the m largest positive eigenvalues. The adequacy of the resulting solution might be assessed using one of the following two criteria

$$P_m^{(1)} = \frac{\sum_{i=1}^{m} |\lambda_i|}{\sum_{i=1}^{n} |\lambda_i|}$$

$$P_m^{(2)} = \frac{\sum_{i=1}^{m} \lambda_i^2}{\sum_{i=1}^{n} \lambda_i^2}$$

Again we would look for values above 0.8 to claim a "good" fit.

Alternatively, there are two criteria for deciding on the number of dimensions for the spatial model to adequately represent the observed proximities:

Trace criterion: Choose the number of coordinates so that the sum of the positive eigenvalues is approximately equal to the sum of all the eigenvalues.

Magnitude criterion: Accept as genuinely positive only those eigenvalues whose magnitude substantially exceeds that of the largest negative eigenvalue.

If, however, the matrix $\boldsymbol{B}$ has a considerable number of large negative eigenvalues, classical scaling of the proximity matrix may be inadvisable and some other methods of scaling, for example non-metric scaling (see the next section), might be better employed.

Hypothetical Example

For our first example we will use the small set of multivariate data $\boldsymbol{X}$

```
X = matrix(c(3, 4, 4, 6, 1, 5, 1, 1, 7, 3, 6, 2, 0, 2, 6, 1, 1, 1, 0, 3, 4, 7, 3, 6, 2, 2, 2, 5, 1, 0, 0, 4, 1, 1,
1, 0, 6, 4, 3, 5, 7, 6, 5, 1, 4, 2, 1, 4, 3, 1), 10, byrow = T)
```

```
     [,1] [,2] [,3] [,4] [,5]
```

```
 [1,]    3    4    4    6    1
 [2,]    5    1    1    7    3
 [3,]    6    2    0    2    6
 [4,]    1    1    1    0    3
 [5,]    4    7    3    6    2
 [6,]    2    2    5    1    0
 [7,]    0    4    1    1    1
 [8,]    0    6    4    3    5
 [9,]    7    6    5    1    4
[10,]    2    1    4    3    1
```

and the associated matrix of Euclidean distances (computed via the dist() function) will be our proximity matrix

```
> (D <- dist(X))
```

```
          1          2          3          4          5          6          7          8          9
2   6.557439
3   6.557439  5.477226
4   8.124038  6.244998  5.744563
5   4.242641  7.937254  6.855655  8.124038
6   7.615773  7.681146  3.605551  4.472136  6.782330
7  10.246951  8.000000  7.211103  8.185353  7.681146  7.280110
8   7.416198  4.242641  4.898979  6.708204  6.557439  6.708204  4.690416
9   9.273618  7.681146  4.582576  6.164414  8.831761  4.000000  7.141428  7.416198
10  9.273618  8.774964  7.141428  7.874008  6.480741  6.164414  3.605551  6.557439 5.830952
```

To apply classical scaling to this matrix in R, we can use the cmdscale() function to do the scaling:

```
> cmdscale(D, k = 9, eig = TRUE)
```

```
$points
            [,1]        [,2]       [,3]       [,4]        [,5]          [,6]
 [1,] -1.6038325 -2.38060903  2.2301092 -0.3656856  0.11536476  8.421598e-08
 [2,] -2.8246377  2.30937202  3.9523782  0.3419185  0.33169405 -2.312861e-08
 [3,] -1.6908272  5.13970089 -1.2880306  0.6503227 -0.05133897  1.074339e-08
 [4,]  3.9527719  2.43233961 -0.3833746  0.6863995 -0.03460933  3.254229e-08
 [5,] -3.5984894 -2.75538195  0.2551393  1.0783741 -1.26125237 -2.345830e-08
 [6,]  2.9520356 -1.35475175  0.1899027 -2.8211220  0.12385813 -3.319300e-08
 [7,]  3.4689928 -0.76411068 -0.3016531  1.6369166 -1.94209512  6.148806e-09
 [8,]  0.3545235 -2.31408566 -2.2161772  2.9240116  2.00450379 -6.654403e-09
 [9,] -2.9362323  0.01279597 -4.3117385 -2.5122743 -0.18911558  1.773836e-08
[10,]  1.9256952 -0.32526941  1.8734445 -1.6188611  0.90299062  3.093294e-09
              [,7]
 [1,] 0.000000e+00
 [2,] 1.565503e-08
 [3,] 1.532955e-08
 [4,] 2.353591e-08
 [5,] 3.588394e-08
 [6,] 6.457705e-09
 [7,] 1.530651e-08
 [8,] 1.652824e-08
 [9,] 2.040312e-08
```

```
[10,] 4.576136e-08

$eig
 [1]  7.518716e+01  5.880560e+01  4.960516e+01  3.042789e+01  1.037419e+01
 [6]  1.086006e-14  5.381235e-15 -5.316641e-15 -8.539862e-15 -1.050854e-14

$x
NULL

$ac
[1] 0

$GOF
[1] 1 1

Warning message:
In cmdscale(D, k = 9, eig = TRUE) :
  only 7 of the first 9 eigenvalues are > 0
```

Note that as $q = 5$ in this example, eigenvalues six to nine are essentially zero and only the first five columns of points represent the Euclidean distance matrix. First we should confirm that the five-dimensional solution achieves complete recovery of the observed distance matrix. We can do this simply by comparing the original distances with those calculated from the five-dimensional scaling solution coordinates using the following R code:

```
> max(abs(dist(X) - dist(cmdscale(D, k = 5))))

[1] 1.154632e-14
```

This confirms that all the differences are essentially zero and that therefore the observed distance matrix is recovered by the five-dimensional classical scaling solution.

We can also check the duality of classical scaling of Euclidean distances and principal components analysis mentioned previously in the chapter by comparing the coordinates of the five-dimensional scaling solution given above with the first five principal component (up to signs) scores obtained by applying PCA to the covariance matrix of the original data; the necessary R code is

```
> max(abs(prcomp(X)$x) - abs(cmdscale(D, k = 5)))

[1] 2.961346e-14
```

Now let us look at two examples involving distances that are not Euclidean. First, we will calculate the Manhattan distances between the rows of the small data matrix $\boldsymbol{X}$. The Manhattan distance for units i and j is given by

$$\sum_{k=1}^{q} \left| x_{ik} - x_{jk} \right|$$

and these distances are not Euclidean.

The R code for calculating the Manhattan distances and then applying classical multidimensional scaling to the resulting distance matrix is:

```
X_m <- cmdscale(dist(X, method = "manhattan"), k = nrow(X) - 1, eig = TRUE)
```

The criteria $P_m^{(1)}$ and $P_m^{(2)}$ can be computed from the eigenvalues as follows:

```
> (X_eigen <- X_m$eig)

[1]  2.806843e+02  2.494246e+02  2.288549e+02  9.250710e+01  4.250504e+01
[6]  2.196808e+01 -2.131628e-14 -1.507023e+01 -2.804630e+01 -5.682752e+01
```

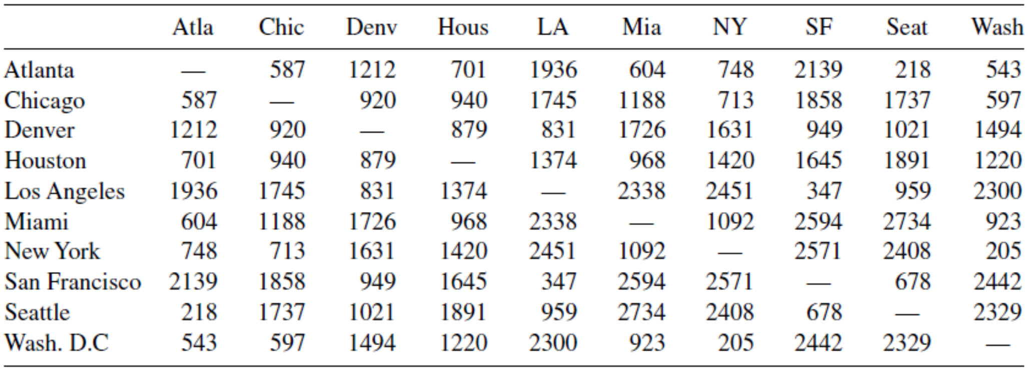
Note that some of the eigenvalues are negative in this case.

```
> cumsum(abs(X_eigen)) / sum(abs(X_eigen))

[1]0.2762945 0.5218182 0.7470939 0.8381542 0.8799945 0.9016190 0.9016190 0.9164536
[9]0.9440612 1.0000000

> cumsum(X_eigen^2) / sum(X_eigen^2)

[1]0.3779304 0.6763685 0.9276127 0.9686639 0.9773307 0.9796457 0.9796457 0.9807352
[9]0.9845085 1.0000000
```

The values of both criteria suggest that a three-dimensional solution seems to fit well.

Airline Distances

For our second example of applying classical multidimensional scaling to non-Euclidean distances, we shall use the airline distances between ten US cities given in Table 4.1.



	Atla	Chic	Denv	Hous	LA	Mia	NY	SF	Seat	Wash
Atlanta	—	587	1212	701	1936	604	748	2139	218	543
Chicago	587	—	920	940	1745	1188	713	1858	1737	597
Denver	1212	920	—	879	831	1726	1631	949	1021	1494
Houston	701	940	879	—	1374	968	1420	1645	1891	1220
Los Angeles	1936	1745	831	1374	—	2338	2451	347	959	2300
Miami	604	1188	1726	968	2338	—	1092	2594	2734	923
New York	748	713	1631	1420	2451	1092	—	2571	2408	205
San Francisco	2139	1858	949	1645	347	2594	2571	—	678	2442
Seattle	218	1737	1021	1891	959	2734	2408	678	—	2329
Wash. D.C	543	597	1494	1220	2300	923	205	2442	2329	—

Table 4.1: airdist data. Airline distances between ten US cities.

Codes to produce Table 4.1 are

```
airdist = matrix(c(0, 587, 1212, 701, 1936, 604, 748, 2139, 218, 543,
                   587, 0, 920, 940, 1745, 1188, 713, 1858, 1737, 597,
                 1212, 920, 0, 879, 831, 1726, 1631, 949, 1021, 1494,
                  701, 940, 879, 0, 1374, 968, 1420, 1645, 1891, 1220,
                 1936, 1745, 831, 1374, 0, 2338, 2451, 347, 959, 2300,
                 604, 1188, 1726, 968, 2338, 0, 1092, 2594, 2734, 923,
                 748, 713, 1631, 1420, 2451, 1092, 0, 2571, 2408, 205,
                 2139, 1858, 949, 1645, 347, 2594, 2571, 0, 678, 2442,
                 218, 1737, 1021, 1891, 959, 2734, 2408, 678, 0, 2329,
                  543, 597, 1494, 1220, 2300, 923, 205, 2442, 2329, 0),10,byrow = T)
colnames(airdist) = c("Atla", "Chic", "Denv", "Hous", "LA", "Mia", "NY", "SF", "Seat", "Wash")
rownames(airdist) = c("Atlanta", "Chicago", "Denver", "Houston", "Los Angeles", "Miami", "New York",
"San Francisco", "Seattle", "Wash. D.C")
```

These distances are not Euclidean since they relate essentially to journeys along the surface of a sphere. To apply classical scaling to these distances and to see the eigenvalues, we can use the following R code:

```
> airline_mds <- cmdscale(airdist, k = 9, eig = TRUE)
```

```
Warning message:
In cmdscale(airdist, k = 9, eig = TRUE) :
  only 6 of the first 9 eigenvalues are > 0
```

```
> airline_mds$points
```

```
                   [,1]       [,2]       [,3]        [,4]         [,5]         [,6]
Atlanta       -434.7569  724.11547  441.15092   0.20613855  -0.03021692 -0.004984303
Chicago       -412.6330   55.22019 -370.94316   3.26472364   8.81904102 -8.803683650
Denver         468.2107 -180.48755 -213.83348  31.23858687  -1.39061684  8.247308287
Houston       -175.5855 -515.15008  362.48515  13.71529882 -18.13320797 -3.462175686
Los Angeles   1206.4323 -466.00611   57.04182  -3.64861784  17.29308372  1.212733385
Miami        -1161.4249 -478.43757  479.52742 -15.57039188   8.69240052  1.890859778
New York     -1115.6133  199.96947 -429.50744 -28.93835970  -6.21554774  5.403971008
San Francisco 1422.7440 -308.47287 -205.97864 -23.28411298 -10.58463038 -2.426079687
Seattle       1221.5707  887.19156  170.62603  -0.07464122   0.01331095  0.005474065
Wash. D.C    -1018.9440   82.05749 -290.56861  23.09137574   1.53638364 -2.063423196
```

The eigenvalues are

```
> (lam <- airline_mds$eig)
[1]  9.212993e+06  2.200398e+06  1.082981e+06  3.343205e+03  9.361601e+02
[6]  2.019022e+02 -5.977574e-09 -2.147508e+03 -1.014155e+04 -1.723147e+06
```

As expected (as the distances are not Euclidean), some of the eigenvalues are negative and so we will again use the criteria $P_m^{(1)}$ and $P_m^{(2)}$ to assess how many coordinates we need to adequately represent the observed distance matrix. The values of the two criteria calculated from the eigenvalues are

```
> cumsum(abs(lam)) / sum(abs(lam))
```

```
[1]0.6471485 0.8017111 0.8777829 0.8780178 0.8780835 0.8780977 0.8780977 0.8782486
[9]0.8789609 1.0000000
```

```
> cumsum(lam^2) / sum(lam^2)
[1]0.9042867 0.9558698 0.9683651 0.9683652 0.9683652 0.9683652 0.9683652 0.9683652
[9]0.9683663 1.0000000
```

These values suggest that the first two coordinates will give an adequate representation of the observed distances. The scatterplot of the two-dimensional coordinate values is shown in Figure 4.1 using the following code

```
> plot(airline_mds$points[,1],airline_mds$points[,2],
    type="n",xlab="Coordinate 1",ylab="Coordinate 2",
    xlim=c(-1500,1500), ylim=c(-1500,1500))
> text(airline_mds$points[,1],airline_mds$points[,2],
    labels=colnames(airdist))
```

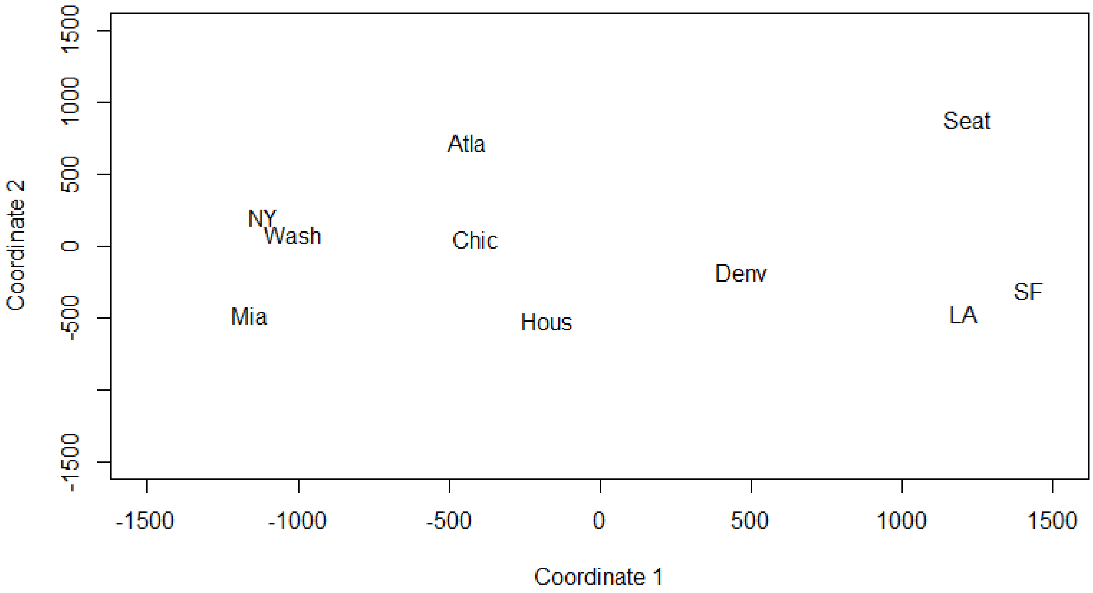


Fig. 4.1. Two-dimensional classical MDS solution for airline distances. The known spatial arrangement is clearly visible in the plot.

In this two-dimensional representation, the geographical location of the cities has been very well recovered by the two-dimensional multidimensional scaling solution obtained from the airline distances.

3. Non-metric multidimensional scaling

In some psychological work and in market research, proximity matrices arise from asking human subjects to make judgments about the similarity or dissimilarity of objects or stimuli of interest. When collecting such data, the investigator may feel that realistically subjects are only able to give "ordinal" judgments; for example, when comparing a range of colors they might be able to specify with some confidence that one color is brighter than another but would be far less confident if asked to put a value to how much brighter.

Such considerations led, in the 1960s, to the search for a method of multidimensional scaling that uses only the rank order of the proximities to produce a spatial representation of them. In other words, a method was sought that would be invariant under monotonic transformations of the observed proximity matrix; i.e., the derived coordinates will remain the same if the numerical values of the observed proximities are changed but their rank order is not.

The quintessential component of the method is that the coordinates in the spatial representation of the observed proximities give rise to fitted distances, d_{ij} , and that these distances are related to a set of numbers which we will call *disparities*, $\hat{d}_{ij}$, by the formula

$$d_{ij} = \hat{d}_{ij} + \epsilon_{ij}$$

where the ϵ_{ij} are error terms representing errors of measurement plus distortion errors arising because the distances do not correspond to a configuration in the particular number of dimensions chosen. The disparities are monotonic with the observed proximities and, subject to this constraint, resemble the fitted distances as closely as possible. In general, only a weak monotonicity constraint is applied, so that if, say, the observed dissimilarities, δ_{ij} are ranked from lowest to highest to give

$$\delta_{i_1 j_1} < \delta_{i_2 j_2} < \cdots < \delta_{i_N j_N}$$

where $N = \frac{1}{2}n(n-1)$, then

$$\hat{d}_{i_1 j_1} < \hat{d}_{i_2 j_2} < \cdots < \hat{d}_{i_N j_N}$$

Monotonic regression is used to find the disparities, and then the required coordinates in the spatial representation of the observed dissimilarities, which we denote by $\widehat{\boldsymbol{X}}(n \times m)$, are found by minimizing a criterion, S, known as Stress, which is a function of $\widehat{\boldsymbol{X}}(n \times m)$ and is defined as

$$S(\hat{X}) = \min \frac{\sum_{i<j} \left(\hat{d}_{ij} - d_{ij}\right)^2}{\sum_{i<j} d_{ij}^2}$$

where the minimum is taken over $\hat{d}_{ij}$ such that $\hat{d}_{ij}$ is monotonic with the observed dissimilarities. In essence, Stress represents the extent to which the rank order of the fitted distances disagrees with the rank order of the observed dissimilarities. The denominator in the formula for Stress is chosen to make the final spatial representation invariant under changes of scale; i.e., uniform stretching or shrinking. For each value of the number of dimensions, m, in the spatial configuration, the configuration that has the smallest Stress is called the best-fitting configuration in m dimensions, S_m, and a rule of thumb for judging the fit is $S_m \geq 20\%$, poor, $S_m = 10\%$, fair, $S_m \leq 5\%$, good; and $S_m = 0$, perfect (this only occurs if the rank order of the fitted distances matches the rank order of the observed dissimilarities and event, which is, of course, very rare in practice).

Judgements of World War II leaders

In this example we shall use the method to get a spatial representation of the judgments of the dissimilarities in ideology of a number of world leaders and politicians prominent at the time of the Second World War, shown in Table 4.5.

	Htl	Mss	Chr	Esn	Stl	Att	Frn	DGl	MT	Trm	Chm	Tit
Hitler	0	3	4	7	3	8	3	4	8	9	4	7
Mussolini	3	0	6	8	5	9	2	4	9	9	5	8
Churchill	4	6	0	4	6	3	5	3	8	5	5	2
Eisenhower	7	8	4	0	8	8	7	5	9	4	4	4
Stalin	3	5	6	8	0	9	6	6	6	7	7	7
Attlee	8	9	3	9	8	0	7	5	9	8	2	8
Franco	3	2	5	7	6	7	0	4	8	8	2	3
De Gaulle	4	4	3	5	6	5	4	0	7	4	5	2
Mao Tse-Tung	8	9	8	9	6	9	8	7	0	4	9	4
Truman	9	9	5	4	7	8	8	4	4	0	5	5
Chamberlin	4	5	5	4	7	2	2	5	9	5	0	7
Tito	7	8	2	4	7	8	3	2	4	5	7	0

Table 4.5: WWIIleaders data. Subjective distances between WWII leaders.

R code to produce Table 4.5:

```
Htl = c(0, 3, 4, 7, 3, 8, 3, 4, 8, 9, 4, 7)
Mss = c(3, 0, 6, 8, 5, 9, 2, 4, 9, 9, 5, 8)
Chr = c(4, 6, 0, 4, 6, 3, 5, 3, 8, 5, 5, 2)
Esn = c(7, 8, 4, 0, 8, 9, 7, 5, 9, 4, 4, 4)
Stl = c(3, 5, 6, 8, 0, 8, 6, 6, 6, 7, 7, 7)
Att = c(8, 9, 3, 8, 9, 0, 7, 5, 9, 8, 2, 8)
Frn = c(3, 2, 5, 7, 6, 7, 0, 4, 8, 8, 2, 3)
DGl = c(4, 4, 3, 5, 6, 5, 4, 0, 7, 4, 5, 2)
MT =  c(8, 9, 8, 9, 6, 9, 8, 7, 0, 4, 9, 4)
Trm = c(9, 9, 5, 4, 7, 8, 8, 4, 4, 0, 5, 5)
Chm = c(4, 5, 5, 4, 7, 2, 2, 5, 9, 5, 0, 7)
Tit = c(7, 8, 2, 4, 7, 8, 3, 2, 4, 5, 7, 0)
WWIIleaders = cbind(Htl, Mss, Chr, Esn, Stl, Att, Frn, DGl, MT, Trm, Chm, Tit)
rownames(WWIIleaders) = c('Hitler', 'Mussolini', 'Churchill', 'Eisenhower', 'Stalin', 'Attlee', 'Franco',
'De Gaulle', 'Mao Tse-Tung', 'Truman', 'Chamberlin', 'Tito')
```

The subject made judgments on a nine-point scale, with the anchor points of the scale, 1 and 9, being described as indicating “very similar” and “very dissimilar”, respectively; this was all the subject was told about the scale.

The non-metric multidimensional scaling applied to these distances is

```
> (WWII_mds <- isoMDS(WWIIleaders))
```

```
initial  value 20.400713
iter   5 value 15.219311
iter   5 value 15.207733
iter   5 value 15.207733
final    value 15.207733
converged
$points
                   [,1]       [,2]
Hitler       -2.6142309 -1.7379740
Mussolini    -3.9069970 -1.2052278
Churchill     0.3240108  1.4415523
Eisenhower    3.0235774  2.8142761
Stalin       -1.4970418 -3.7542676
Attlee       -2.0361086  5.0863312
Franco       -2.8641212  0.1038147
De Gaulle     0.6474253 -0.2385760
Mao Tse-Tung  4.0870170 -4.6631954
Truman        4.4895837  0.2057644
Chamberlin   -2.1052548  2.7707306
Tito          2.3601946 -1.0613770

$stress
[1] 15.20773
```

The two-dimensional solution appears in Figure 4.7 by the following code

```
> plot(WWII_mds$points[,1], WWII_mds$points[,2],
     type="n",xlab="Coordinate 1",ylab="Coordinate 2",
     xlim=c(-6,6), ylim=c(-6,6))
> text(WWII_mds$points[,1], WWII_mds$points[,2],
     labels=rownames(WWIIleaders))
```

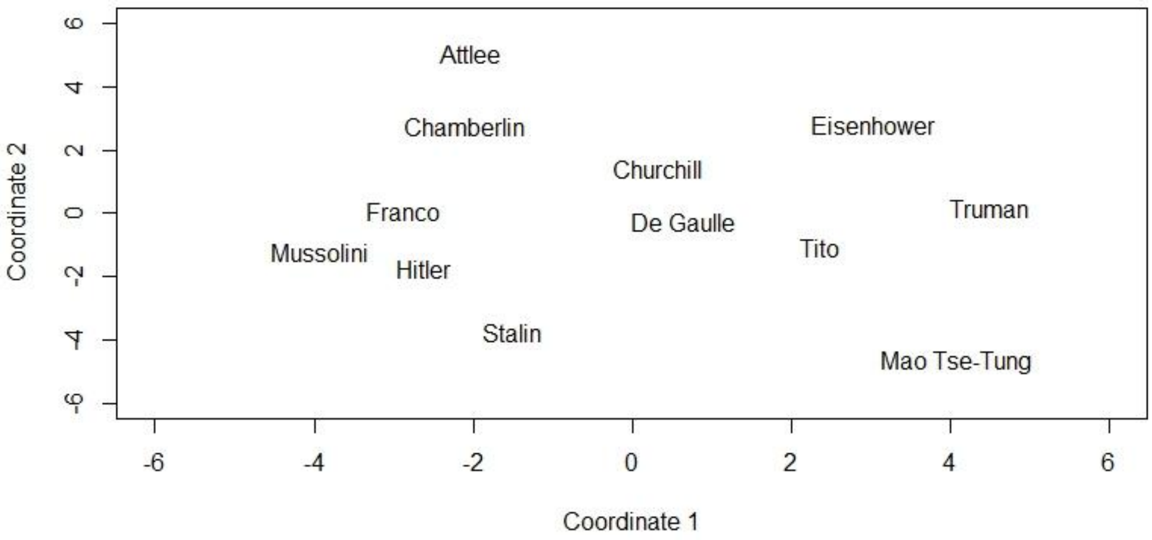


Fig. 4.7. Non-metric multidimensional scaling of perceived distances of World War II leaders.

Clearly, the three fascists group together (Hitler-Mussolini-Franco) as do the three British prime ministers(Attlee-Chur-Cham). Stalin and Mao Tse-Tung are more isolated compared with the other leaders. Eisenhower seems more related to the British government than to his own President Truman. Interestingly, de Gaulle is placed in the center of the MDS solution.

4. Correspondence analysis

Correspondence analysis, a form of multidimensional scaling, is essentially an approach to construct a spatial model that displays the associations among a set of categorical variables.
Mathematically, correspondence analysis can be regarded as either

- a method for decomposing the chi-squared statistic used to test for independence in a contingency table into components corresponding to different dimensions of the heterogeneity between its columns, or
- a method for simultaneously assigning a scale to rows and a separate scale to columns so as to maximize the correlation between the two scales.

Quintessentially, however, correspondence analysis is a technique for displaying multivariate (most often bivariate) categorical data graphically by deriving coordinates to represent the categories of both the row and column variables, which may then be plotted so as to display the pattern of association between the variables graphically. Here we give accounts of the method demonstrating the use of classical multidimensional scaling to get a two-dimensional map to represent a set of data in the form of a two-dimensional contingency table. The general two-dimensional contingency table in which there are r rows and c columns can be written as

using an obvious dot notation for summing the counts in the contingency table over rows or over columns. From this table we can construct tables of column proportions and row proportions given by

Column proportions $p_{ij}^{c} = \frac{ij}{}$,

Row proportions $p_{ij}^{r} = \frac{ij}{}$,

What is known as the chi-squared distance between columns i and j is defined as

$$d_{ij}^{(\text{cols})} = \sum_{k=1}^{r} \frac{1}{p_{k.}} \left(p_{ki}^{c} - p_{kj}^{c}\right)^2,$$

where $p_{k.} = \frac{n_{k.}}{n}$.

The chi-square distance is seen to be a weighted Euclidean distance based on column proportions. It will be zero if the two columns have the same values for these proportions. It can also be seen from the weighting factors, $\frac{1}{p_{k.}}$, that rare categories of the column variable have a greater influence on the distance than common ones. A similar distance measure can be defined for rows i and j as

$$d_{ij}^{(\text{rows})} = \sum_{k=1}^{c} \frac{1}{p_{.k}} \left(p_{ik}^{r} - p_{jk}^{r}\right)^2$$

where $p_{.k} = \frac{n_{.k}}{n}$.

A correspondence analysis "map" of the data can be found by applying classical MDS to each distance matrix in turn and plotting usually the first two coordinates for column categories and those for row categories on the same diagram, suitably labeled to differentiate the points representing row categories from those representing column categories. The resulting diagram is interpreted by examining the positions of the points representing the row categories and the column categories. The relative values of the coordinates of these points reflect associations between the categories of the row variable and the categories of the column variable. Assuming that a two-dimensional solution provides an adequate fit for the data, row points that are close together represent row categories that have similar profiles (conditional distributions) across columns. Column points that are close together indicate columns with similar profiles (conditional distributions) down the rows. Finally, row points that lie close to column points represent a row/column combination that occurs more frequently in the table than would be expected if the row and column variables were independent. Conversely, row and column points that are distant from one another indicate a cell in the table where the count is lower than would be expected under independence.

Smoking and Motherhood

Table 4.8 shows a set of frequency data showing the distribution of birth outcomes by age of mother, length of gestation, and whether or not the mother smoked during the prenatal period. We shall consider the data as a two-dimensional contingency table with four row categories and four column categories.

	Premature		**Full term**	
	Died in 1st year (pd)	Alive at year 1 (pa)	Died in 1st year (ftd)	Alive at year 1 (fta)
Young mothers				
Nonsmokers (YN)	50	315	24	4012
Smokers (YS)	9	40	6	459
Old mothers				
Nonsmokers (ON)	41	147	14	1594
Smokers (OS)	4	11	1	124

Table 4.8 Smoking and Motherhood

R Codes to produce Table 4.8

```
SM = matrix(c(50, 315, 24, 4012, 9, 40 ,6, 459, 41, 147, 14, 1594, 4, 11, 1, 124), 4, 4, byrow=T)
rownames(SM) = c('YN', 'YS', 'ON', 'OS')
colnames(SM) = c('pd', 'pa', 'ftd', 'fta')
```

The obvious question of interest for the data in Table 4.8 is whether or not a mother's smoking puts a newborn baby at risk. However, several other questions might also be of interest. Are smokers more likely to have premature babies? Are older mothers more likely to have premature babies? And how does smoking affect premature babies?

The calculation of the two-dimensional classical multidimensional scaling solution based on the row- and column-wise chi-squared distance measure can be computed via cmdscale(); however, we first have to compute the necessary row and column distance matrices, and we will do this by setting up a small convenience function as follows:

```
> D <- function(x) {
                a <- t(t(x) / colSums(x))
                ret <- sqrt(colSums((a[,rep(1:ncol(x), ncol(x))] -
                a[, rep(1:ncol(x), rep(ncol(x), ncol(x)))])^2 *
                sum(x) / rowSums(x)))
                matrix(ret, ncol = ncol(x))
                      }
> (dcols <- D(SM))
          [,1]      [,2]      [,3]      [,4]
[1,] 0.0000000 0.2950077 0.2697176 0.3654828
[2,] 0.2950077 0.0000000 0.2309099 0.0729345
[3,] 0.2697176 0.2309099 0.0000000 0.2799938
[4,] 0.3654828 0.0729345 0.2799938 0.0000000

> (drows <- D(t(SM)))
           [,1]       [,2]       [,3]       [,4]
[1,] 0.00000000 0.09635833 0.10746773 0.14609151
[2,] 0.09635833 0.00000000 0.06635467 0.10607244
[3,] 0.10746773 0.06635467 0.00000000 0.04883099
[4,] 0.14609151 0.10607244 0.04883099 0.00000000
```

Applying classical MDS to each of these distance matrices gives the required two-dimensional coordinates with which to construct our "map" of the data. Plotting those with suitable labels and with the axes suitably scaled to reflect the greater variation on dimension one than on dimension two is achieved using the R code presented with Figure 4.8.

```
> r1 <- cmdscale(dcols, eig = TRUE)
> c1 <- cmdscale(drows, eig = TRUE)
> plot(r1$points, xlim = range(r1$points[,1], c1$points[,1]) * 1.5,
     ylim = range(r1$points[,1], c1$points[,1]) * 1.5, type = "n",
     xlab = "Coordinate 1", ylab = "Coordinate 2", lwd = 2)
> text(r1$points, labels = colnames(SM), cex = 0.7)
> text(c1$points, labels = rownames(SM), cex = 0.7)
> abline(h = 0, lty = 2)
> abline(v = 0, lty = 2)
```

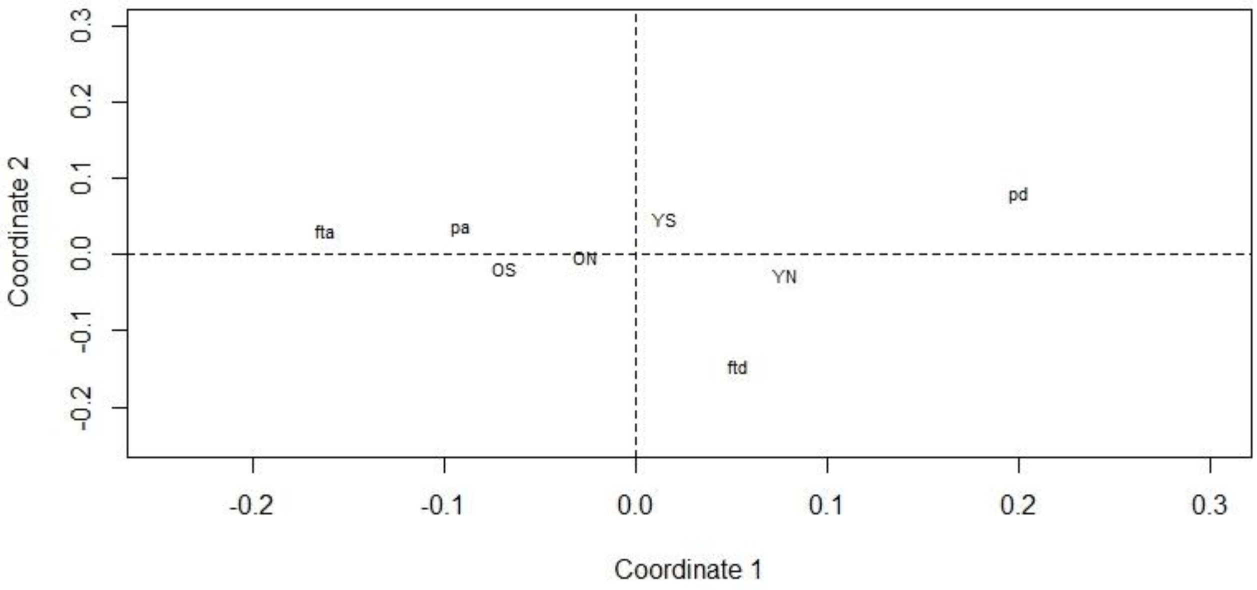


Fig. 4.8. Correspondence analysis for smoking and motherhood data.

This diagram suggests that young mothers who smoke tend to have more full-term babies who then die in their first year, and older mothers who smoke have rather more than expected premature babies who die in the first year. It does appear that smoking is a risk factor for death in the first year of the baby's life, and that age is associated with length of gestation, with older mothers delivering more premature babies.

Hodgkin's Disease

	Response			
Historical type	Positive	Partial	None	Total
LP	74	18	12	104
NS	68	16	12	96
MC	154	54	58	266
LD	18	10	44	72
Total	314	98	126	538

Table 4.9 Hodgkin's Disease

The data shown in Table 4.9 were recorded during a study of Hodgkin's disease, a cancer of the lymph nodes. Each of 538 patients with the disease was classified by histological type, and by their response to treatment three months after it had begun. The histological classification is:

- lymphocyte predominance (LP),
- nodular sclerosis (NS),
- mixed cellularity (MC),
- lymphocyte depletion (LD).

The key question is, “What, if any, is the relationship between histological type and response to treatment?”

In this example the two-dimensional solution from applying classical MDS to the chi-squared distances gives a perfect fit. The resulting scatterplot is shown in Figure 4.9.

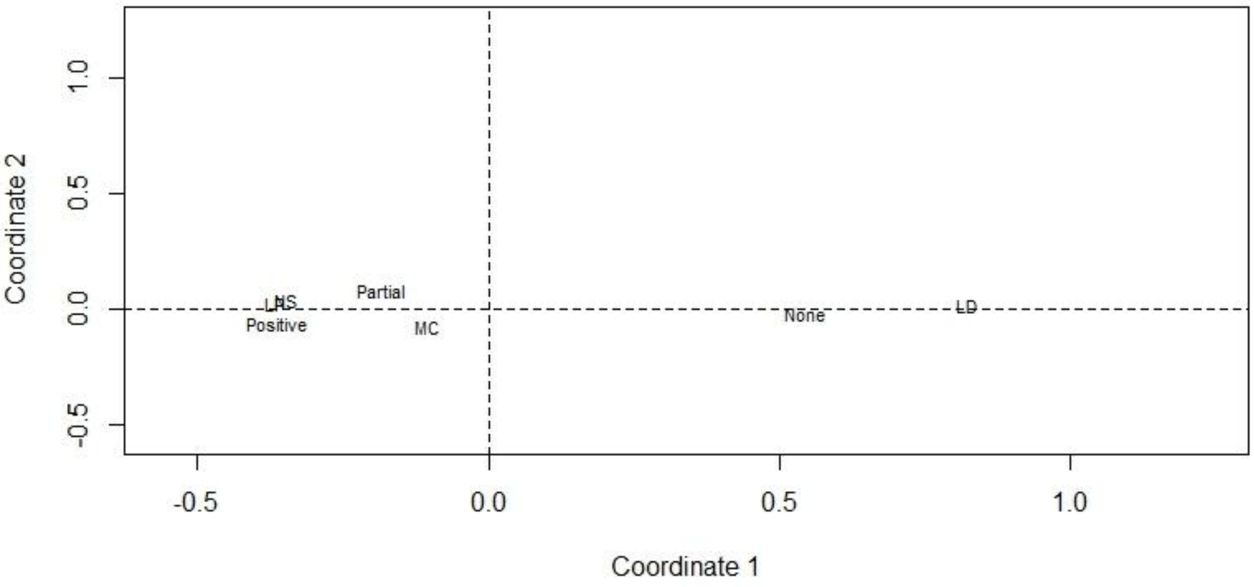


Figure 4.9 Classical MDS two-dimensional solution for Hodgkin’s disease data.

The positions of the points representing histological classification and response to treatment in this diagram imply the following:

- Lymphocyte depletion tends to result in no response to treatment.
- Nodular sclerosis and lymphocyte predominance are associated with a positive response to treatment.
- Mixed cellularity tends to result in a partial response to treatment.