Correspondence Analysis & Related Methods

Michael Greenacre

SESSION 3:
MULTIDIMENSIONAL SCALING (MDS)
DIMENSION REDUCTION
CLASSICAL MDS
NONMETRIC MDS

Distances and dissimilarities...

- $n$ objects
- $d_{ij} =$ distance between object $i$ and object $j$

Properties of a distance (metric)
1. $d_{ij} = d_{ji}$
2. $d_{ij} \geq 0, \quad d_{ij} = 0 \iff i = j$
3. $d_{ij} \leq d_{ik} + d_{kj}$ (the triangle inequality)

(If 3. not satisfied we often talk of a dissimilarity)

The chi-square distance is a true distance, whereas Bray-Curtis is a dissimilarity

Distances and maps...

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Amsterdam</td>
<td>0</td>
<td>2979</td>
<td>1533</td>
<td>768</td>
<td>676</td>
<td>1076</td>
</tr>
<tr>
<td>Athens</td>
<td>2979</td>
<td>0</td>
<td>3261</td>
<td>2594</td>
<td>2486</td>
<td>3250</td>
</tr>
<tr>
<td>Barcelona</td>
<td>1533</td>
<td>3261</td>
<td>0</td>
<td>1061</td>
<td>1945</td>
<td>600</td>
</tr>
<tr>
<td>Basel</td>
<td>768</td>
<td>2594</td>
<td>1061</td>
<td>0</td>
<td>884</td>
<td>898</td>
</tr>
<tr>
<td>Berlin</td>
<td>676</td>
<td>2486</td>
<td>1945</td>
<td>884</td>
<td>0</td>
<td>1631</td>
</tr>
<tr>
<td>Bordeaux</td>
<td>1076</td>
<td>3250</td>
<td>600</td>
<td>898</td>
<td>1631</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td>:</td>
<td>:</td>
<td>:</td>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
</tbody>
</table>

Observed distances $d_{ij}$

OK

Fitted distances

Multidimensional scaling (MDS)

CITIES

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Amsterdam</td>
<td>0</td>
<td>$d_{12}$</td>
<td>$d_{13}$</td>
<td>$d_{14}$</td>
<td>$d_{15}$</td>
<td>$d_{16}$</td>
</tr>
<tr>
<td>Athens</td>
<td>$d_{21}$</td>
<td>0</td>
<td>$d_{23}$</td>
<td>$d_{24}$</td>
<td>$d_{25}$</td>
<td>$d_{26}$</td>
</tr>
<tr>
<td>Barcelona</td>
<td>$d_{31}$</td>
<td>$d_{32}$</td>
<td>0</td>
<td>$d_{34}$</td>
<td>$d_{35}$</td>
<td>$d_{36}$</td>
</tr>
<tr>
<td>Basel</td>
<td>$d_{41}$</td>
<td>$d_{42}$</td>
<td>$d_{43}$</td>
<td>0</td>
<td>$d_{45}$</td>
<td>$d_{46}$</td>
</tr>
<tr>
<td>Berlin</td>
<td>$d_{51}$</td>
<td>$d_{52}$</td>
<td>$d_{53}$</td>
<td>$d_{54}$</td>
<td>0</td>
<td>$d_{56}$</td>
</tr>
<tr>
<td>Bordeaux</td>
<td>$d_{61}$</td>
<td>$d_{62}$</td>
<td>$d_{63}$</td>
<td>$d_{64}$</td>
<td>$d_{65}$</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td>:</td>
<td>:</td>
<td>:</td>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
</tbody>
</table>

Fitted distances $\hat{d}_{ij}$

etc...
**Multidimensional scaling (MDS)**

Objective is to minimize some measure of discrepancy, or error, between observed and fitted distances.

- **Observed distances**
  \[ d_{ij} \]

- **Minimize** \( \sum_{ij} (d_{ij} - \hat{d}_{ij})^2 \)
  or
  \[ \text{also called “Sammon’s non-linear mapping”; R function \texttt{sammon}} \]

- **Fitted distances**
  \[ \hat{d}_{ij} \]

- **Minimize** \( \sum_{ij} (f(d_{ij}) - \hat{d}_{ij})^2 \)
  for any monotonically increasing function \( f \)
  or
  \[ \text{for any nonmetric MDS, similar idea to that of Spearman’s rank correlation; R function \texttt{isoMDS}} \]

Maximize the agreement between the rank-ordered distances in the map and the rank-ordering of the original distances (nonmetric MDS), similar idea to that of Spearman’s rank correlation; R function \texttt{isoMDS}.

**“Classical” MDS**

Fits the distances indirectly.

Classical (“YoHoToGo”*) MDS situates the points in a space of as high dimensionality as possible to reproduce the observed distances and then projects the points onto low-dimensional suspaces, usually a plane:

**Metric and nonmetric MDS**

These methods fit the interpoint distances directly

**Stress**: measures the discrepancy between the observed distances (data) and the fitted distances (map)

- **Observed distances**
  \[ d_{ij} \]

  Raw stress: \( \sum_{ij} (d_{ij} - \hat{d}_{ij})^2 \)

  Normalized stress: \( \frac{\sum_{ij} (d_{ij} - \hat{d}_{ij})^2}{\sum_{ij} d_{ij}^2} \)

- **Fitted distances**
  \[ \hat{d}_{ij} \]

  Kruskal stress: \( \sqrt{\frac{\sum_{ij} (d_{ij} - \hat{d}_{ij})^2}{\sum_{ij} \hat{d}_{ij}^2}} \)

  used in R function \texttt{isoMDS} for nonmetric MDS, can be thought of as a percentage error

**MDS of Bray-Curtis dissimilarities - classical**

**Stress**: measures the discrepancy between the observed distances (data) and the fitted distances (map)

- **Observed distances**
  \[ d_{ij} \]

  Raw stress: \( \sum_{ij} (d_{ij} - \hat{d}_{ij})^2 \)

  Normalized stress: \( \frac{\sum_{ij} (d_{ij} - \hat{d}_{ij})^2}{\sum_{ij} d_{ij}^2} \)

  Kruskal stress: \( \sqrt{\frac{\sum_{ij} (d_{ij} - \hat{d}_{ij})^2}{\sum_{ij} \hat{d}_{ij}^2}} \)

  used in R function \texttt{isoMDS} for nonmetric MDS, can be thought of as a percentage error

**Goodness of fit**: 53.1%
MDS of Bray-Curtis dissimilarities – nonmetric

Stress: 13.5%

MDS of chi-square distances – classical

Goodness of fit: 74.4%

Correspondence analysis

Notice that the rows and the columns are depicted in a joint map. To be continued...

Goodness of fit: 75.2%
Correspondence Analysis &
Related Methods

Michael Greenacre

SESSION 4:
CLASSICAL MDS – the computations

In this course we concentrate on the STRUCTURAL methods of multivariate analysis

- methods that reveal continuous structures (scales, dimensions, factors...)
- methods that reveal discrete structures (clusters, groups, segments, partitions...)

factorial methods

- multidimensional scaling (MDS)
- principal components analysis (PCA)
- factor analysis (FA)
- correspondence analysis (CA)

metric MDS
- non-metric MDS

cluster analysis
- hierarchical clustering
- non-hierarchical clustering

Basic concept: distance

Classical scaling

- From a map to a distance matrix

points  distances

(3,4) \* 2
(-1,3) \* 1
(3,2) \* 3
(-1,-1) \* 4

(squared) distance matrix

\[
\begin{pmatrix}
0 & 17 & 17 & 16 \\
17 & 0 & 4 & 41 \\
16 & 4 & 0 & 25 \\
16 & 41 & 25 & 0 \\
\end{pmatrix}
\]

• suppose you have \( n \) points \( x_i \) \( (i=1,...,n) \) in \( p \)-dimensional Euclidean space

\[
X = \begin{bmatrix}
x_{11} & x_{12} & \cdots & x_{1p} \\
x_{21} & x_{22} & \cdots & x_{2p} \\
\vdots & \vdots & \ddots & \vdots \\
x_{n1} & x_{n2} & \cdots & x_{np}
\end{bmatrix}
\]

- squared distance between the \( i \)-th and \( j \)-th points is

\[
\delta_{ij} = \sum_{k=1}^{p} (x_{ik} - x_{jk})^2
\]

\( \Delta = \begin{bmatrix}
\delta_{11} & \delta_{12} & \cdots & \delta_{1n} \\
\delta_{21} & \delta_{22} & \cdots & \delta_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
\delta_{n1} & \delta_{n2} & \cdots & \delta_{nn}
\end{bmatrix}
\]
**Classical scaling**

- in matrix notation:

\[
\Delta = \begin{bmatrix}
\delta_{11} & \delta_{12} & \cdots & \delta_{1n} \\
\delta_{21} & \delta_{22} & \cdots & \delta_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
\delta_{n1} & \delta_{n2} & \cdots & \delta_{nn}
\end{bmatrix} = s1^T + 1s^T - 2S
\]

where \( S = XX^T \) and \( s = \text{diag}(S) \) is matrix of scalar products.

- the problem in classical scaling:
  - distances \( \rightarrow \) points

- given \( \Delta \) solve for \( X \)

**R code to double-centre and eigendecompose**

```r
# read in the squared distance matrix
d2 <- matrix(c(0,17,17,16,17,0,4,41,17,4,0,25,16,4,1,25,0), nrow=4)

# compute scalar products
n <- nrow(d2)
oones <- rep(1, n)
I <- diag(oones)
Sd <- -0.5 * (I - (1/n) * oones %*% t(oones)) %*% d2 %*% (I - (1/n) * oones %*% t(oones))

# compute eigenvalues and eigenvectors using R function eigen
Sd.eig <- eigen(Sd)

# compute coordinates and plot
X <- Sd.eig$vectors[,1:2] %*% sqrt(Sd.eig$values[1:2])
plot(X, type="n")
text(X, labels=1:4)
```

**Classical scaling**

- if we had \( S \) and had to recover \( X \) it would be simple:

\[
S = XX^T
\]

- recall the eigenvalue-eigenvector decomposition of a square symmetric matrix, for example of \( S \):

\[
S = U\Lambda U^T
\]

where

\[
UU^T = I; \quad \Lambda = \begin{bmatrix}
\lambda_1 & 0 & \cdots & 0 \\
0 & \lambda_2 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \lambda_n
\end{bmatrix} \quad \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0
\]

so a possible solution would be:

\[
X = U\Lambda^{1/2}
\]

**Classical scaling**

- but we don’t have the scalar products \( S \) but rather the squared distances \( \Delta = s1^T + 1s^T - 2S \)

- we can recover the matrix of scalar products \( S^* \) with respect to the centroid of the \( n \) points by a transformation of \( \Delta \) called double-centring:
  - subtract the row means from all the squared distances
  - subtract column means from the resultant matrix
  then multiply double-centred matrix by \(-1/2\) to obtain \( S^* \)

Then carry on as before:

\[
S^* = U\Lambda U^T
\]

\[
X^* = U\Lambda^{1/2}
\]