Multidimensional Scaling in R: Comprehensive Guide

Dani

1 Introduction to Multidimensional Scaling (MDS)

Multidimensional Scaling (MDS) is a set of methods for visualizing the structure of distance data in a lower-dimensional space. The purpose is to project high-dimensional dissimilarities into a 2D or 3D plane while retaining as much information about their relative similarities as possible.

1.1 Key Types of MDS

There are two primary types of MDS based on the treatment of dissimilarities:

- **Metric MDS**: Preserves the actual dissimilarity values, making it suitable for data where the numerical distance has a direct interpretation.
- **Non-Metric MDS**: Focuses on preserving the rank order of dissimilarities rather than exact values. This is useful for data where only the order of similarity matters (e.g., preference rankings).

1.2 Choosing Between Metric and Non-Metric MDS

The decision between Metric and Non-Metric MDS often depends on whether the distance matrix contains meaningful magnitudes. For example:

- Use **Metric MDS** when:
  - You have continuous data with interpretable Euclidean distances.
  - Exact distances matter more than their order.
- Use **Non-Metric MDS** when:
  - Dealing with ordinal or categorical data.
  - Only the relative ranking of distances is meaningful.
  - You suspect that a non-linear relationship might better capture the patterns in the data.

2 Distance Calculations using the `cluster` Package

2.1 Understanding the `daisy()` Function

- The `daisy()` function is highly versatile for computing distance matrices in R, particularly for mixed-type data.
- **Available Distance Metrics**:
  - **Euclidean Distance**: Measures straight-line distance and assumes continuous, numeric data.
  - **Gower Distance**: Handles mixed data types (numeric, ordinal, and categorical).
- **Example**:

```
library(cluster)
data("iris")

# Calculate Euclidean distance
euclidean_dist <- daisy(iris[, -5], metric = "euclidean")
```

```r
# Calculate Gower distance
gower_dist <- daisy(iris[, -5], metric = "gower")
```

- **Practical Consideration**:
  - `daisy()` is often preferable over `dist()` when working with data frames containing non-numeric variables.
  - Consider scaling your data when using Euclidean distances to avoid variables with larger scales dominating the result.

3 Principal Coordinates Analysis using `labdsv`

- **Principal Coordinates Analysis (PCoA)** is a linear method that represents dissimilarities in Euclidean space.
- Ideal for visualizing data with a known distance matrix.
- Example using the `eurodist` dataset:

```r
library("labdsv")

# Perform Principal Coordinates Analysis (PCoA)
pco_eu <- pco(eurodist, k = 2)

# Visualization
plot(pco_eu$points[,1], pco_eu$points[,2], type = "n", main = "PCoA
    of European Cities")
text(pco_eu$points[,1], pco_eu$points[,2], labels(eurodist), cex =
    0.8)
```

4 Classical MDS using `cmdscale()` from the `stats` Package

- Classical MDS ('cmdscale') is ideal when the goal is to retain the true Euclidean distances.
- Computes a configuration that best fits the input dissimilarity matrix.
- Example:

```r
# Perform Classical MDS on eurodist dataset
mds_eu <- cmdscale(eurodist, eig = TRUE)

# Plotting the MDS solution
plot(mds_eu$points[,1], mds_eu$points[,2], main = "Classical MDS",
    type = "n")
text(mds_eu$points[,1], mds_eu$points[,2], labels(eurodist), cex =
    0.8)
```

- **Interpretation**:
  - The first two components typically capture the majority of the variance.
  - The eigenvalues indicate the variance explained by each dimension.
  - Negative eigenvalues suggest poor fit and indicate that the distance matrix might not be truly Euclidean.

5 Non-Metric MDS using `isoMDS()` from the `MASS` Package

- Non-Metric MDS ('isoMDS') preserves the rank order of dissimilarities.
- Useful for understanding non-linear relationships in the data.

```
library(MASS)

# Perform Non-Metric MDS
usa_nmds <- isoMDS(dist(USArrests))

# Visualization
plot(usa_nmds$points, main = "Non-Metric MDS on USArrests", type =
    "n")
text(usa_nmds$points, labels = row.names(USArrests), cex = 0.8)
```

- **Interpreting Stress Values**:
  - The 'stress' value indicates how well the 2D configuration matches the original dissimilarities.
  - Lower stress values (< 0.1) indicate a good fit, while higher values suggest distortions.

6 Non-Linear Mapping with `sammon()`

- The `sammon()` function in R performs a form of non-metric MDS, but it emphasizes preserving local structures.

```
nmds_sam <- sammon(dist(USArrests))
plot(nmds_sam$points, main = "Sammon Mapping on USArrests", type =
    "n")
text(nmds_sam$points, labels = row.names(USArrests))
```

- **When to Use Sammon Mapping**:
  - When maintaining local neighborhood distances is more important than preserving global structure.

7 3D Visualizations using the `rgl` Package

- For visualizing MDS results in 3D, use the 'rgl' package:

```
library(rgl)

# 3D Visualization
plot3d(mds_eu$points[,1:3], main = "3D MDS Visualization of
    European Cities")
```

- **Practical Tip**: Use 'rgl' to interactively rotate and explore the 3D configuration.

8 Advanced Considerations: Goodness-of-Fit Analysis

- Use the sum of squared eigenvalues and stress values to evaluate how well the MDS solution represents the original data.
- Example:

```
sum(mds_eu$eig[1:2]) / sum(mds_eu$eig)  # Proportion of variance
    explained by first 2 dimensions
```

- **Key Insight**: A higher proportion suggests that the first two dimensions capture most of the variance, making the 2D plot a good representation.