## Introduction

Multidimensional scaling (MDS) is a technique used in multivariate statistics to represent similarities between high-dimensional data. This is done by using the distance between points as a dissimilarity measure, with the aim of being able to display the relative distances between all points in reduced dimensions.

The technique of MDS will be demonstrated by analysing a real dataset. The dataset that will be used is of protein expression in control and Down syndrome mice with different exposures to a drug and learning methods. This data will be analysed using variations of MDS to present how the methodology affects results.

## Purpose of MDS

The primary purpose of MDS is to arrange points representing objects in a way so that the geometrical distance between points reflects relationships between the objects. This is done through two objectives, reducing dimensionality and visual representation.

The first main objective is to reduce the dimensionality of data. This is significant in multivariate statistics, as datasets with large numbers of variables and observations can be difficult to interpret without a high level of analysis. MDS addresses this by working with the dissimilarities between data, which can reduce the overall number of dimensions needed to represent the analysis. The dissimilarity data (represented by distances in space) can be scaled to find a lower dimensional configuration that maintains pairwise distances as well as possible.

The second main objective of MDS is as a method of visually representing the similarity or dissimilarity between data points. The results of MDS analysis can be graphed in two- or three-dimensional space so that observations that are more similar are visually closer together. This is a very intuitive and interpretable way of representing similarities between observations. Visually representing these similarities can help to identify overall patterns or clusters in the data.

MDS is similar to principal component analysis (PCA), as both are tools that can be used to visualise the relative distance between data. The primary difference is that MDS uses the similarity of the data whereas PCA analyses the distance between observations. In PCA, the principal components are obtained through eigen-decomposition of a correlation, covariance or cross-product matrix. It is not possible to perform PCA on a distance/similarity matrix, as it they are not positive semi-definite matrices. MDS transforms such a matrix into an equivalent cross-product matrix which can then be analysed similar to PCA.

A basic example of the application of MDS is a problem such as that presented by Krabbe (2017). In this problem, the aim is to construct a map of airport locations only knowing the distances between all the airports. With only a few airports this can be simple to deduce by hand, but as the number of airports and therefore distances increases, the problem becomes more complex. With MDS these relative distances can be interpreted to almost perfect locations of the airports. This can also be done using only the ranking of the distances by magnitude, as shown in Figure 1 below.

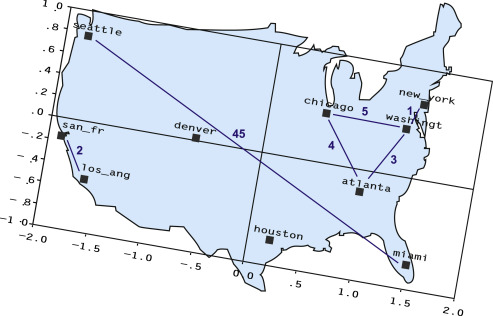


Figure 1 - The two-dimensional solution of multidimensional scaling on the ranks of the 45 airline mileages (with distance ranks 1–5 and 45 depicted) between 10 cities in the United States (Krabbe, 2017)

## Measuring similarity

The input to MDS analysis is a matrix that indicates relationships among a set of items. This is most commonly represented as proximity data given by a proximity matrix **D** that represents the dissimilarity between objects. The way this proximity matrix is calculated can significantly affect the analysis. There are various ways to create the proximity matrix **D**, that depend on the structure of the data used. Examples of ways of obtaining proximity data include:

* Euclidean distance – the straight-line distance between points in space
* Human evaluated similarity – using data based on people’s perceived similarity of objects
* Confusion data – the frequency at which objects are categorised to be the same, or how objects are “confused” with another

In the analysis for this report, Euclidean distance will be used. For two different data points Xi and Xj in R-dimensional space, the distance between Xi and Xj is defined as:

d(xi, xj) =

## Types of scaling

To scale an object is to change its dimensions by a certain factor. In MDS, this scaling is done to reduce the dimensionality of the data and map the objects 1 toN to embedding points x1,…,xN. These embedding points are configured in a way so that the given similarity values Di,j are closely approximated by the distances .

There are several different types of scaling that can be used in MDS, which differ primarily in the loss function used to scale the proximity data. While there are many scaling options, this report will focus on three common types:

* Classical multidimensional scaling, also known as Principle Coordinate Analysis(PCoA);
* Non-metric multidimensional scaling (nMDS);
* Metric multidimensional scaling (mMDS);

In PCoA, the coordinate matrix of embedding points **X** is obtained by eigenvalue decomposition based on . The matrix **B** is computed based on the proximity matrix using the following steps:

1. Establish the squared proximity matrix
2. Transform into matrix B by applying double centering , where J is the centering matrix given by .
3. Select the number of dimensions desired in the output (m), and determine the m largest eigenvalues and the corresponding eigenvectors of B
4. Set Em as the matrix of the m eigenvectors and as the diagonal matrix of the m eigenvalues.

As a result of this, is well approximated by as due to the double centering.

PCoA minimises a loss function called strain, given by:

The strain essentially measures the fit between the inner products and the inner product data Bij.

Non-metric multidimensional scaling is a group of MDS techniques where the embedding points X are found based on the ranks of the pairwise distances. This is done through isotonic regression Because of this, nonmetric MDS is robust to non-linear relationships between the calculated dissimilarities and the projected distance between objects.

Metric multidimensional scaling is another group of MDS techniques that uses the actual values of the distances. A type of metric MDS that will be explored in this report is weighted metric MDS. In this version of MDS, weighting of observations and variables is considered in obtaining the coordinate matrix of the observations. Suppose we have a dataset of *n* observations and *m* variables; the weights are computed as follows

1. Define δ as a vector of length ½ n(n-1) of given squared observed distances. This could also be described as one half of the symmetric matrix of distances strung out as a vector.
2. Define **,** meaning X is the matrix of squared pairwise differences between observation by each variable. This matrix will have the dimensions (½ n(n-1)) x m .
3. Fit the multiple regression model *δ = Xw + e* , for which the solution is defined as

*w = (XTX)-1δ*

In doing this, the weights are calculated to minimise the least-squares difference between the observed dissimilarities/distances and the calculated weighted distances/dissimilarities. Once the weights have been calculated, the embedding points of the observations can be found through generalised singular value decomposition. This is done as follows:

1. Pre-process the data matrix Y by pre- and post-multiplying it by the square roots of Dr and Dw, which are the diagonal matrices of observation and variable weights respectively. This can also be written as
2. Perform singular value decomposition on the pre-processed data matrix
3. Obtain the matrix of principal coordinates (F) for observations using
4. Obtain the matrix of standard coordinates (G) for variables using
5. The coordinates for a two-dimensional solution would be the linear combination of the first two rows of F and G

In this process, the following function is minimised:

where r­I is the weighting of observation i, and ŷi is the closest low-dimensional approximation of observation yi. This function is called inertia and measures the difference between the original and the approximated matrix.

## Data used

The mice protein expression dataset that will be analysed is based on the protein expression measured in control and Down syndrome (trisomy) mice. The data was collected to understand the effects of treatments on the mice’s learning abilities, which were measured through the expression levels of gene proteins that are believed to contribute to learning ability. The mice also underwent various treatments; administering a drug called memantine or a saline placebo, and administering an electrical shock after being exposed to the novel learning environment (context-shock) or an electrical shock straight away (shock-context). The experiment that this dataset is sourced from took 15 measurements of each protein per mouse. As there were 72 mice, this means that the dataset is comprised of 72x15=1080 observations.

In total, 82 variables were available in the dataset:

* 1: Mouse ID
* 2-78: Values of expression levels of 77 proteins; the names of proteins are followed by \_n indicating that they were measured in the nuclear fraction. For example: DYRK1A\_n
* 79: Genotype: control (c) or trisomy (t)
* 80: Treatment type: memantine (m) or saline (s)
* 81: Behaviour: context-shock (CS) or shock-context (SC)
* 82: Class: c-CS-s, c-CS-m, c-SC-s, c-SC-m, t-CS-s, t-CS-m, t-SC-s, t-SC-m

The variables of interest are the protein expression levels and the class.

To decrease the amount of data to be analysed, only one observation of the 15 for each mouse was used. This meant that 72 observations were analysed. To make observations easy to identify and discuss, they were coded 1-72.

## Methodology

The R software package was used to analyse the data according to the methodologies described above. For the principal coordinate analysis, the base-R function *cmdscale* was used.

## Results

Euclidean distance was used to calculate the pairwise distances between the mice. Some of the results for this can be seen in Table 1 below.

*Table 1: Example of calculated Euclidean distances between some of the observations in the dataset*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 2 | 1.6121691 |  |  |  |  |  |  |
| 3 | 3.3762983 | 3.4316124 |  |  |  |  |  |
| 4 | 2.5819008 | 2.6245166 | 1.4972049 |  |  |  |  |
| 5 | 2.6071157 | 2.4223115 | 2.6029592 | 1.5032000 |  |  |  |
| 6 | 1.4990678 | 1.8776086 | 3.6283388 | 2.8236700 | 2.1901033 |  |  |
| 7 | 1.4465051 | 2.1676145 | 4.1052513 | 3.0560403 | 2.4820914 | 1.2253682 |  |
| 8 | 1.6383400 | 1.4208126 | 3.0150684 | 1.8015819 | 1.7584210 | 2.1575546 | 2.0191255 |

### Classical multidimensional scaling (PCoA)

PCoA was used to reconfigure the data into two dimensions. The results of this can be seen in Figure 1 below, where the observations have also been colour-coded by their class.

A screenshot of a social media post

Description automatically generated

Figure : Visualisation of results of PCoA analysis on 72 mice from the protein expression dataset

As we can see in Figure 1, there are no strong clusters formed by the various classes of mice. Interestingly, mouse 25 is very far away from the other data points, suggesting its protein expression levels were very different to other mice. This mouse also had the largest calculated pairwise Euclidean distance from another observation, of approximately 7 whereas most distances ranged from 0 to 5. This suggests that the scaling used in the PCoA approximated this observation’s distance relatively well.

Based on the eigenvalues calculated in the PCoA, the first scaled dimension was able to account for 42.1% of the variation in the data, the second dimension accounts for 27.9%, and the third dimension 13.6%. We can therefore conclude that when reducing this dataset to do dimension we can still preserve 70% of the variation in the data.

## References

Data reference

<https://www.sciencedirect.com/science/article/pii/B9780128015049000143>