NON-METRIC MULTIDIMENSIONAL SCALING (MDS)

Steven M. Holland

Department of Geology, University of Georgia, Athens, GA 30602-2501

May 2008

Introduction

Nonmetric multidimensional scaling (MDS, also NMDS and NMS) is an ordination technique that differs in several ways from nearly all other ordination methods. In most ordination methods, many axes are calculated, but only a few are viewed, owing to graphical limitations. In MDS, a small number of axes are explicitly chosen prior to the analysis and the data are fitted to those dimensions; there are no hidden axes of variation. Second, most other ordination methods are analytical and therefore result in a single unique solution to a set of data. In contrast, MDS is a numerical technique that iteratively seeks a solution and stops computation when an acceptable solution has been found, or it stops after some pre-specified number of attempts. As a result, an MDS ordination is not a unique solution and a subsequent MDS analysis on the same set of data and following the same methodology will likely result in a somewhat different ordination. Third, MDS is not an eigenvalue-eigenvector technique like principal components analysis or correspondence analysis that ordinates the data such that axis 1 explains the greatest amount of variance, axis 2 explains the next greatest amount of variance, and so on. As a result, an MDS ordination can be rotated, inverted, or centered to any desired configuration.

Unlike other ordination methods, MDS makes few assumptions about the nature of the data. For example, principal components analysis assumes linear relationships and reciprocal averaging assumes modal relationships. MDS makes neither of these assumptions, so is well suited for a wide variety of data. MDS also allows the use of any distance measure of the samples, unlike other methods which specify particular measures, such as covariance or correlation in PCA or the implied chi-squared measure in detrended correspondence analysis.

MDS does suffer from two principal drawbacks, although these are becoming less important as computational power increases. First, MDS is slow, particularly for large data sets and the reasons for this will become apparent under Computation below. Second, because MDS is a numerical optimization technique, it can fail to find the true best solution because it can become stuck on local minima, solutions that are not the best solution but that are better than all nearby solutions. Increased computational speed now allows MDS ordinations even of large data sets and allows multiple ordinations to be run, such that the chance of being stuck on a local minimum is greatly decreased.

Computation

The method underlying MDS is straightforward in approach, but computationally demanding to execute. First, one starts with a matrix of data consisting of n rows of samples and p columns of variables, such as taxa for ecological data. From this, a n x n symmetrical matrix of

all pairwise distances among samples is calculated with an appropriate distance measure, such as Euclidean distance, Manhattan distance (city block distance), and Bray distance. The MDS ordination will be performed on this distance matrix.

Next, a desired number of m dimensions is chosen for the ordination. Note that an n-dimensional ordination is not equivalent to the first n dimensions of an n+1-dimensional ordination; the two ordinations would have to be run separately.

The MDS software begins by constructing an initial configuration of the samples in the m dimensions. This initial configuration could be based on another ordination or it could consist of an entirely random arrangement of the samples. The final ordination is partly dependent on this initial configuration, so a variety of approaches are used to avoid the problem of local minima. One approach is to perform several ordinations, each starting from a different random arrangement of points, and to select the ordination with the best fit. Another approach is to perform a different type of ordination, such as a principal components analysis or a higher-order MDS, and to use n axes from that ordination as the initial configuration. A third approach, useful for data thought to be geographically arrayed, is to use the geographic locations of samples as a starting configuration.

Distances among samples in this starting configuration are calculated, typically with a Euclidean metric. These distances are regressed against the original distance matrix and the predicted ordination distances for each pair of samples is calculated. A variety of regression methods can be used, including linear, polynomial, and non-parametric approaches, the last of which stipulates only that the regression consistently increases from left to right. In any case, the regression is fitted by least-squares. In a perfect ordination, all ordinated distances would fall exactly on the regression, that is, they would match the rank-order of distances in the original distance matrix. The goodness of fit of the regression is measured based on the sum of squared differences between ordination-based distances and the distances predicted by the regression. This goodness of fit is called stress and can be calculated in several ways, with one of the most common being Kruskal's Stress (formula 1)

$$Stress(1) = \sqrt{\sum_{h,i} \left(d_{hi} - \hat{d}_{hi} \right)^2 / \sum_{h,i} d_{hi}^2}$$

where d_{hi} is the ordinated distance between samples h and i, and d-hat is the distance predicted from the regression.

This configuration is then improved by moving the positions of samples in ordination space by a small amount in the direction of steepest descent, the direction in which stress changes most rapidly. The ordination distance matrix is recalculated, the regression performed again, and stress recalculated, and this entire procedure of nudging samples and recalculating stress is repeated until some small specified tolerance value is achieved or until the procedure converges by failing to achieve any lower values of stress, which indicates that a minimum (perhaps local) has been found.

Considerations

The ordination will be sensitive to the number of dimensions that is chosen, so this choice must be made with care. Choosing too few dimensions will force multiple axes of variation to be expressed on a single ordination dimension. Choosing too many dimensions is no better in that it can cause a single source of variation to be expressed on more than one dimension. One way to choose an appropriate number of dimensions is perform ordinations of progressively higher numbers of dimensions. A scree diagram (stress versus number of dimensions) can then be plotted, on which one can identify the point beyond which additional dimensions do not substantially lower the stress value. A second criterion for the appropriate number of dimensions is the interpretability of the ordination, that is, whether the results make sense.

The stress value reflects how well the ordination summarizes the observed distances among the samples. Several "rules of thumb" for stress have been proposed, but have been criticized for being over-simplistic. Stress increases both with the number of samples and with the number of variables. For the same underlying data structure, a larger data set will necessarily result in a higher stress value, so use caution when comparing stress among data sets. Stress can also be highly influenced by one or a few poorly fit samples, so it is important to check the contributions to stress among samples in an ordination.

Although MDS seeks to preserve the distance relationships among the samples, it is still necessary to perform any transformations to obtain a meaningful ordination. For example, in ecological data, samples should be standardized by sample size to avoid ordinations that reflect primarily sample size, which is generally not of interest.

MDS in R

R has two main MDS functions available, isoMDS, which is part of the MASS library, and metaMDS, which is part of the vegan library. The metaMDS routine allows greater automation of the ordination process, so is usually the preferred method. The metaMDS function uses isoMDS in its calculations as well as several helper functions. The metaMDS routine also has the useful default behavior of following the ordination with a rotation via principal components analysis such that MDS axis 1 reflects the principal source of variation, and so on, as is characteristic of eigenvalue methods. Other commercial packages of MDS may not perform such a rotation, so use caution in interpreting their results.

1) Download and install the vegan library, necessary for running the metaMDS() command.

```
> library(vegan)
```

2) Run an MDS on data set, with columns of variables and rows of samples. The vegan package is designed for ecological data, so the metaMDS default settings are set with this in mind. For example, the distance metric defaults to Bray and common ecological data transformations are turned on by default. For non-ecological data, these settings may distort the ordination.

```
> mydata <- read.table("mydata.txt", header=TRUE, row.names=1,
   sep=",")
> mydata.mds <- metaMDS(mydata)
# the default MDS ordination

> mydata.mds.ALT <- metaMDS(mydata, distance="euclidean", k=3,
   trymax=50, autotransform=FALSE)
# Shows how an MDS could be performed on non-ecological data,
# where a euclidean distance metric would be appropriate.  The
# transformations appropriate for ecological data are also
# turned off, so one would need to make any necessary
# transformations prior to calling the metaMDS function.  This
# MDS will be 3-dimensional (k=3), and will use 50 starts from
# random configurations to avoid local minima.
```

3) View items in the list produced by metaMDS.

```
> names(mydata.mds)
# mydata.mds$points: sample scores
# mydata.mds$dims: number of MDS axes or dimensions
# mydata.mds$stress: stress value of final solution
# mydata.mds$data: what was ordinated, including any
#  transformations
# mydata.mds$distance: distance metric used
# mydata.mds$converged: whether solution converged or
#  not (T/F)
# mydata.mds$tries: number of random initial configurations
#  tried
# mydata.mds$species: scores of variables (species / taxa
#  in ecology)
# mydata.mds$call: restates how the function was called
```

4) View the results of the MDS, which will display several elements of the list detailed above, including how the function was called (call), the data set and any transformations used (data), the distance measure (distance), the number of dimensions (dims), the final stress value (stress), whether any convergent solution was achieved (converged), how many random initial configurations were tried (tries), plus whether scores have been scaled, centered, or rotated.

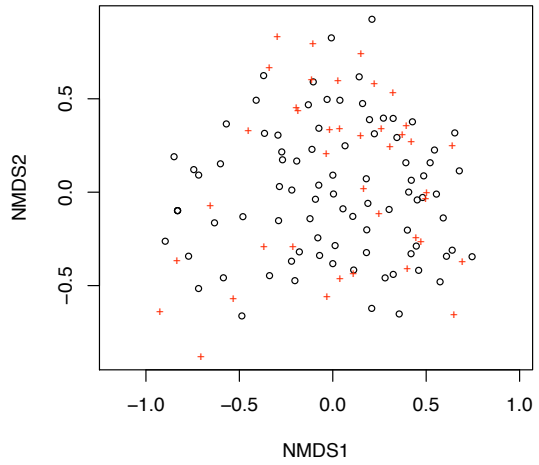
```
> mydata.mds
```

5) Extract sample and variable scores. The column numbers correspond to the MDS axes, so this will return as many columns as was specified with the k parameter in the call to metaMDS.

```
> variableScores <- mydata.mds$species
> sampleScores <- mydata.mds$points
```

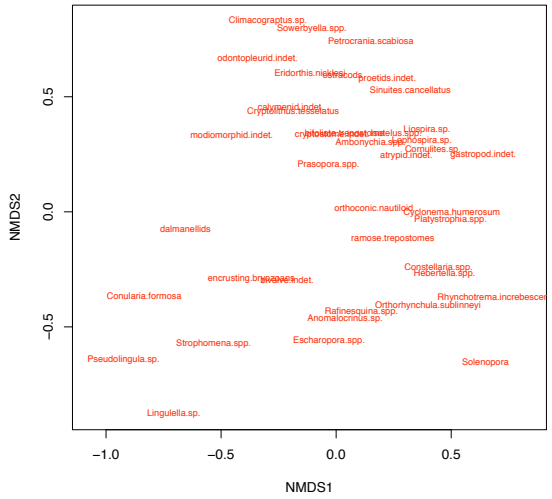
6) Plot sample and variable scores in same space. Open black circles correspond to samples and red crosses indicate taxa.

```
> plot(mydata.mds)
```



7) MDS plots can be customized by selecting either sites or species. Also, labels may be displayed instead of symbols by specifying type="t".

```
> plot(mydata.mds, type="t", display=c("species"))
# text labels instead of symbols
# Crowding of text labels can be alleviated by plotting to
# a larger window
```

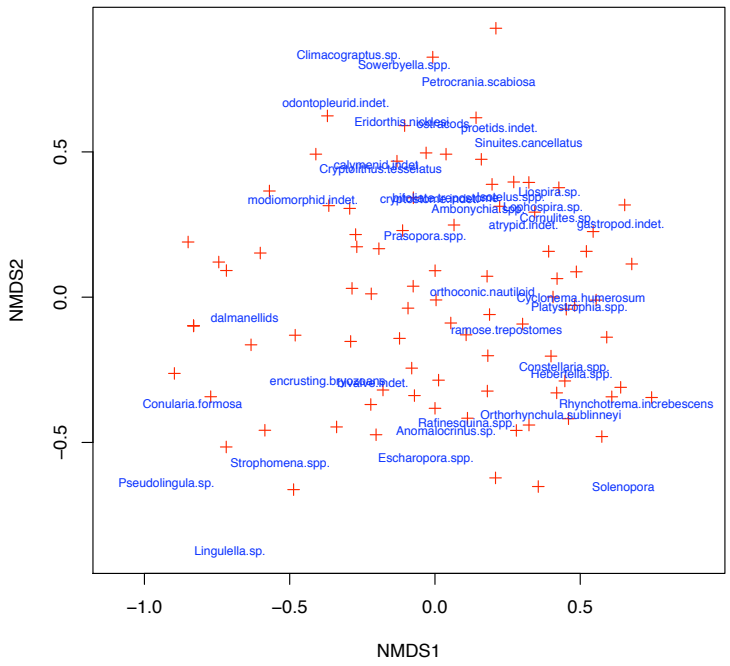


8) MDS plots can be further customized. By specifying type ="n", no sample scores or variable scores will be plotted. These can then be plotted with the points() and text() commands. For crowded plots, congestion of points and labels can be alleviated by plotting to a larger window and by using cex to reduce the size of symbols and text.

```
> plot(mydata.mds, type="n")
# plots axes, but no symbols or labels

> points(mydata.mds, display=c("sites"), choices=c(1,2),
#  pch=3, col="red")
# plots points for all samples (specified by "sites") for MDS
# axes 1 & 2 (specified by choices).  Note that symbols can
# be modified with typical adjustments, such as pch and col.
# See the select parameter on help page for metaMDS for how to
# use a vector to specify which points are plotted.

> text(mydata.mds, display=c("species"), choices=c(1,2),
   col="blue", cex=0.7)
# plots labels for all variable scores (specified by
# "species") for MDS axes 1 & 2.  Typical plotting parameters
# can also be set, such as using cex to plot smaller labels.
# See labels parameter on help page for metaMDS for how to
# specify an alternative vector of labels, as opposed to the
# row or column names, which are used by default.  Note that
# there is also a select parameter for text, as there is for
# points, mentioned above.
```



9) For congested plots, this will display a plot with symbols for both samples and variables. You can then click on individual points you would like identified. Press "escape" when you are done selecting your points to see their identities.

```
> fig <- ordiplot(mydata.mds)
> identify(fig, "spec")
```

References

Legendre, P., and L. Legendre, 1998. Numerical Ecology. Elsevier: Amsterdam, 853 p.

McCune, B., and J.B. Grace, 2002. Analysis of Ecological Communities. MjM Software Design: Gleneden Beach, Oregon, 300 p..