NON-METRIC MULTIDIMENSIONAL SCALING (NMS)

Steven M. Holland

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

May 2008
revised December 2019

Introduction

Nonmetric multidimensional scaling (NMS, also abbreviated NMDS and MDS) is an ordination technique that differs in five important ways from nearly all other ordination methods.

1) Most ordination methods calculate many axes, but they display only a few of those for reasons of practicality. In contrast, NMS calculates only a limited number of axes that are explicitly chosen prior to the analysis. As a result, there are no hidden axes of variation.

2) Most ordination methods are analytical and therefore result in a single unique solution to a set of data. In contrast, NMS 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 NMS ordination is not a unique solution; a subsequent NMS of the same data will likely result in a somewhat different ordination.

3) NMS is not an eigenanalysis technique like principal components analysis or correspondence analysis. As a result, the axes cannot be interpreted 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 NMS ordination can be rotated, inverted, or centered to any desired configuration.

4) NMS makes few assumptions about the nature of the data. For example, principal components analysis assumes linear relationships, and reciprocal averaging assumes modal relationships. NMS makes neither of these assumptions, making it well-suited for a wide variety of data.

5) NMS allows the use of any distance measure, unlike other methods which specify particular measures, such as covariance or correlation in PCA or the implied chi-squared measure in correspondence analysis.

Although NMS is a highly flexible and widely applicable technique, it suffers from two principal drawbacks. First, NMS is slow, particularly for large data sets. Second, because NMS is a numerical optimization technique, it can fail to find the true best solution because it can become stuck on local minima, that is, solutions that are not the best solution but that are better than all nearby solutions. Increasing computational speed is solving both of these problems: large data sets can now be run relatively quickly, and multiple restarts can be used to lessen the chances of a solution remaining on a local minimum.

Computation

The method underlying NMS is straightforward, but computationally demanding. First, it starts with a matrix of data consisting of n rows of samples and p columns of variables (species or taxa in ecological data). There should be no missing values: every variable should have a value for every sample, and this value may be zero. 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), or Bray (Sorenson) distance. The NMS ordination is performed on this distance matrix.

Next, a desired number of k dimensions is chosen for the ordination. The resulting ordination can be greatly sensitive to this number of chosen dimensions. For example, a k-dimensional ordination is not equivalent to the first k dimensions of a k+1-dimensional ordination.

NMS begins by constructing an initial configuration of the samples in the k dimensions. This initial configuration could be based on another ordination or it could consist of an entirely random placement 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 placement 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 NMS, and to use k 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 as 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 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 is performed again, and stress is recalculated. These steps are performed repeatedly 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 is 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. 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. Several guidelines for a "good" value of stress have been proposed, but all have been criticized for being simplistic.

Although NMS seeks to preserve the distance relationships among the samples, it is still necessary to perform any data 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.

NMS in R

R has several NMS functions available. Two are part of the MASS library, so they are automatically installed with R: **`monoMDS()`** and **`isoMDS()`**. Both of these will produce a simple NMS. A better solution is **`metaMDS()`**, which solves several of the problems inherent to NMS; **`metaMDS()`** is part of the **`vegan`** library. The first problem it solves (usually) is the problem of local minima, which it does by restarting the ordination process repeatedly from a different starting configuration, in search of a better solution. The second problem it solves is the arbitrary orientation of the ordinated point cloud, which it does by performing a principal components analysis on the final set of ordinated points. Although this may sound complicated, perhaps even suspicious, it isn't when you remember that PCA is simply a rotation of set of data. In this case, the ordinated points are rotated such that axis 1 now expresses the greatest amount of variation, followed by axis two, etc. The **`metaMDS()`** function automates all of this. By default, it uses the **`monoMDS()`** function to perform the actual ordinations.

It is important to remember that although the ordinated points produced by **`metaMDS()`** have a specific orientation and a specific ordering of axes in terms of explained variance, most NMS implementations do not do this. As a result, the NMS axes from other implementations commonly lack any labeling or scales. This lack of scale or meaninful orientation greatly complicates interpretation of ordinations not produced by `metaMDS()`, so use caution in interpreting their results.

In the code below, R commands are indicated with a `bold monospaced font`. The second and subsequent lines of code of a multiline command are indented. R output is shown in `gray`.

As mentioned, `metaMDS()` is included in the `vegan` package (https://cran.r-project.org/web/packages/vegan), written by Jari Oksanen and colleagues. This package also includes a wide variety of analytic methods for ecological analysis. The first step to using `vegan` is to install the library, a one-time operation, except for installing updates.

1) To run `metaMDS()`, the `vegan` library will need to be loaded with the `library()` function.

```
library(vegan)
```

2) This demonstration will use a data set on Late Ordovician benthic marine macroinvertebrate communities from central Kentucky (Holland and Patzkowsky 2004), available on the GEOL 8370 website (http://strata.uga.edu/8370/data/). The data set consists of three files: the first contains the counts of species within samples, the second contains additional properties of each sample (such as age, depositional environment, etc.), and the third contains the properties of the species (such as phylum, class, order).

```
ky <- read.table('KentuckyCounts.txt', header=TRUE,
  row.names=1, sep=',')

sampleProperties <- read.table('KentuckySampleProperties.txt',
  header=TRUE, row.names=1, sep=',')

speciesProperties <- read.table('KentuckyTaxonProperties.txt',
  header=TRUE, row.names=1, sep=',')
```

3) A two-dimensional NMS can be performed on ecological data using the defaults of `metaMDS()`. These defaults include the Bray (also called Sorenson) distance metric and the Wisconsin double transformation. This data transformation first puts all species on the same scale of abundance to remove the preferential weighting of abundant taxa over rare taxa. Second, it equalizes the total abundance in all samples so that samples with large abundances do not dominate the ordination at the expense of samples with small abundances.

```
kyNMS <- metaMDS(ky)
```

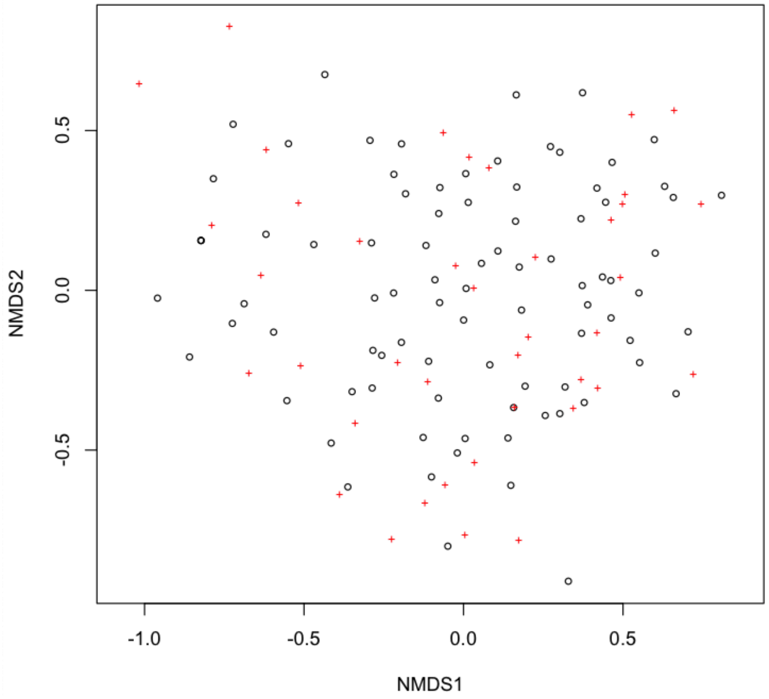
For ecological data, you can also change the number of axes that are calculated by setting the `k` argument to a value higher than the default of 2. If the ordination fails to converge, the number of restarts from a random configuration can be increased from the default of 20, with the only cost being computational time.

4) The object returned belongs to the class “metaMDS”, and the objects within this list can be viewed with **names()** or by displaying the **NMS** object.

Of most use are the sample scores (**points**), the species scores (**species**). The **NMS** object also contains details about the ordination run, including stress (**stress**), the number of dimensions (**dims**), the distance metric used (**distance**), the number of random initial configurations tried (**tries**), the function call (**call**), and whether the ordination converged (**converged**).

5) The first step to interpreting the ordination is to make a biplot which shows the sample scores and species scores in the ordination space, done most simply by calling **plot()** on the ordination object.

```
plot(kyNMS)
```



Open black circles correspond to samples and red crosses indicate taxa, although none are labelled.

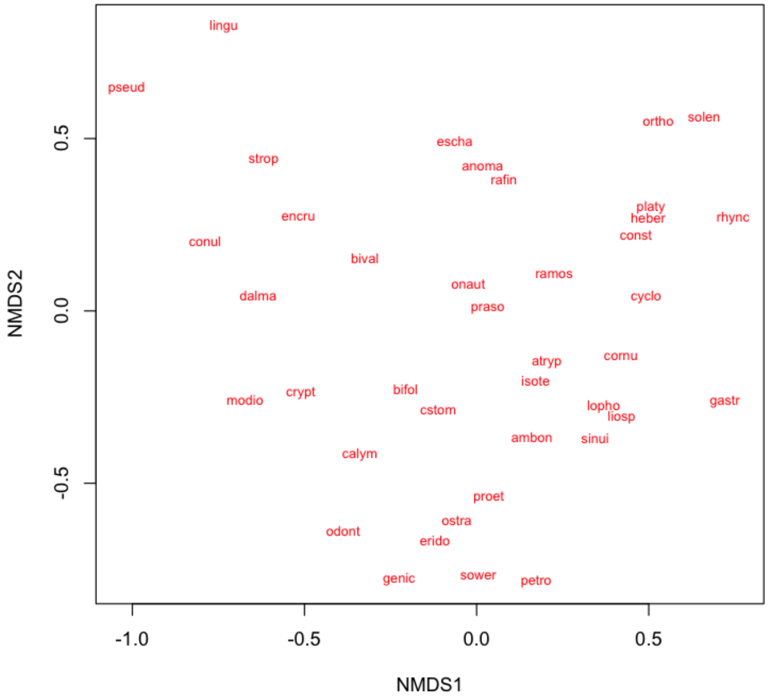
It is possible to create a plot on which individual points (species or samples) can be identified with a click. Press the escape key when you are done selecting points to see their identities.

```
fig <- ordiplot(kyNMS)
identify(fig, 'spec')
```

Because nothing is labelled, the default plot is not very helpful and you will need to customize the plot.

One option is to display only the species or only the site by setting **display**, and by showing the labels for them by specifying **type='t'**. The labels shown correspond to either the row names in the original data (for samples) or the column names, for species. If these labels are long, this plot can get congested, even unreadable from the overlapping labels.

```
plot(kyNMS, type='t', display=c('species'))
```



6) Often, a better option is to use the basic **plot()**, **points()**, and **text()** commands on the extracted scores, which allows more customization as well as avoiding the syntax where **type** and **display** must be set.

The first step is to extract the scores, so that $ notation can be skipped, which is particularly helpful if the name of the NMS object is long.

```
kySpecies <- kyNMS$species
kySamples <- kyNMS$points
```

Interpreting an NMS is best done by plotting the species or samples coded by external data. Samples could be coded, for example, by facies or age. Species could likewise be coded by higher taxa, guild, or life habit. By seeing how such groupings are arranged in ordination space, it becomes possible to understand the meaning of the ordination axes and the sources of variation in community composition.

First, we will examine the sample scores, and this will be done by coding the samples by the facies in which they are found. For convenience and simplicity in the function calls, extract vectors to hold the axis 1 and 2 sample scores. Also, extract a vector holding the lithofacies for each of the samples, which are stored in the **samplePropertie s** data frame created at the beginning. It will also be useful to know how many different facies there are.

```
nms1 <- kySamples[, 1]
nms2 <- kySamples[, 2]
facies <- sampleProperties$Lithofacies
numFacies <- length(levels(facies))
```

To color-code the samples based on the facies they occur in, we find what facies are present using **levels()**. Based on that output, create a vector of colors for each facies (i.e., Deep Subtidal will be blue, Offshore will be black, and so on).

```
levels(facies)
```

```
[1] "Deep Subtidal"    "Offshore"         "Sand Shoal"
"Shallow Subtidal"
```

```
colors <- c('blue', 'black', 'red', 'orange')
```

Finally, create an empty plot and then loop through the facies, adding the points for each, finishing by adding a legend.

```
plot(nms1, nms2, type='n', xlab='NMS 1', ylab='NMS 2',
   las=1, main='Sample scores')

for (i in 1:numFacies) {
  faciesPoints <- facies==levels(facies)[i]
  points(nms1[faciesPoints], nms2[faciesPoints],
    col=colors[i], pch=16)
}

legend(0.3, -0.5, levels(facies), pch=16, col=colors)
```

If you are not comfortable with loops, this could also be done using one of the **apply()**-family of functions or by calling points individually for each facies. Note that because every run of an NMS may be slightly different, such as with a particular axis flipped, the **legend()** coordinates may have to be adjusted to put the legend in a suitable place.

NMS of non-ecological data

NMS can also be performed on non-ecological data, but with four main differences:

First, the default distance metric (bray) is appropriate only for ecological data, in which species show a modal (not linear) response to environmental variables. In most cases with non-ecological data, the distance parameter should be set to euclidean.

Second, the default data transformation that are appropriate for ecological data need to be turned off by setting **autotransform=FALSE** and **noshare=FALSE**.

Third, to calculate variable scores (called species scores), all of your data must be positive. If they are not, you will need to add a constant so that they are.

Fourth, to get variable scores, set **wasscores=TRUE**.

This example will use the Nashville carbonates data set, also used in the principal components lecture, and also available on the GEOL 8370 website. As in the the principal components example, the analysis will be performed only only the geochemical measurements (columns 2–9); the first column contains the stratigraphic position, which could be used as an external variable. These geochemical data are pulled off into their own data frame, and the major element data within them (columns 3–8) are log-transformed to lessen the effect of their long right tails.

```
nashville <- read.table('NashvilleCarbonates.csv',
  header=TRUE, row.names=1, sep=',')
geochem <- nashville[, -1]
geochem[ , 3:8] <- log10(geochem[, 3:8])
```

Because **metaMDS()** needs of the values to be positive (it is set up for ecological data, where a species abundance cannot be negative), the geochemical data must be shifted so that all of their values are positive. The **shiftByMin()** function below will do this, by adding the minimum value for each variable to all values for that variable. The **apply()** function performs **shiftByMin()** on every column in the geochemical data.

```
shiftByMin <- function(x) {x + abs(min(x))}
geochem <- apply(geochem, 2, shiftByMin)
```

At this point, the NMS can be performed on this non-ecological data. This requires setting **distance='euclidean'**, **autotransform=FALSE**, **wasscores=TRUE**, and **noshaare=FALSE**. The number of dimensions (**k**) is set to 3, and the number of restarts (**trymax**) is raised to 50.

```
chemNMS <- metaMDS(geochem, distance='euclidean', k=3,
  trymax=50, autotransform=FALSE, wasscores=TRUE,
  noshare=FALSE)
chemNMS
```

To make a custom of the scores, it is useful to extract the sample scores and variable scores.

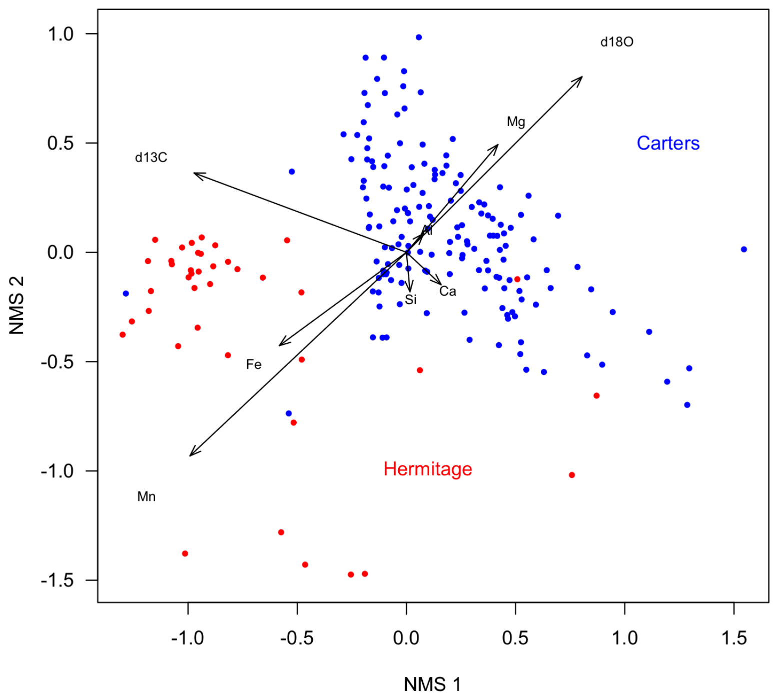
```
chemVars <- chemNMS$species
chemSamples <- chemNMS$points
```

Because a major change in paleoceanography occurs at meter position 34.2 in these data, we will want to color-code our points depending on whether they are below or above this horizon. That horizon also separates the Carters Formation from the overlying Hermitage Formation. To make the subsequent code simpler and less error-prone, two logical vectors are formed to specify the samples corresponding to each formation.

```
Carters <- nashville$StratPosition < 34.2
Hermitage <- nashville$StratPosition > 34.2
```

Make an empty plot, and add the points, color-coded by formation. Also add color-code labels instead of a legend.

```
plot(chemSamples[, 1], chemSamples[, 2], xlab='NMS 1',
  ylab='NMS 2', type='n', asp=1, las=1)
points(chemSamples[Carters, 1], chemSamples[Carters, 2],
  pch=16, cex=0.7, col='blue')
points(chemSamples[Hermitage, 1], chemSamples[Hermitage, 2],
  pch=16, cex=0.7, col='red')
text(1.2, 0.5, 'Carters', col='blue')
text(0.1, -1.0, 'Hermitage', col=‘red')
```



Add vectors for the variables. The scaling factor **`textNudge`** ensures that the labels are plotted slightly before the tips of the arrows.

```
arrows(0, 0, chemVars[, 1], chemVars[, 2], length=0.1,
  angle=20)
textNudge <- 1.2
text(chemVars[, 1]*textNudge, chemVars[, 2]*textNudge,
  rownames(chemVars), cex=0.7)
```

Combining sample scores and variable scores on a plot like this, with the sample scores coded by an external variable, makes for a highly interpretable plot. The Carters Formation carbonates are characterized by large values of d18O and a high Mg/(Mn+Fe) ratio, suggesting dolomitization. In contrast, the Hermitage is characterized by the opposite.

What to report

When reporting a non-metric multidimensional scaling, always include the following:

- A description of any data culling or data transformations that were used before the ordination. State these in the order that they were performed.

- The distance metric, number of random restarts, and the specified number of axes.

- The value of stress.

- A table or plot of variable scores that shows how each variable contributes to each axis of the NMS.

- One or more plots of sample scores that emphasizes the interpretation of the axes, such as color-coding samples by an external variable.

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.