XGvis: Interactive Data Visualization with Multidimensional Scaling

Andreas BUJA [1], Deborah F. SWAYNE [2],
Michael L. LITTMAN [3], Nathaniel DEAN [4],
and Heike HOFMANN [5]

November 29, 2001

We discuss interactive techniques for multidimensional scaling (MDS) and a system, named "XGvis", that implements these techniques.

MDS is a method for visualizing proximity data, that is, data where objects are characterized by dissimilarity values for all pairs of objects. MDS constructs maps of these objects in $\mathbb{R}^k$ by interpreting the dissimilarities as distances.

MDS in its conventional batch implementations is prone to uncertainties with regard to 1) local minima in the underlying optimization, 2) sensitivity to the choice of the optimization criterion, 3) artifacts in point configurations, and 4) local inadequacy of the point configurations. These uncertainties will be addressed by the following interactive techniques: 1) random and manual perturbation of point configurations, 2) interactive control over parameters that determine the criterion and its minimization, 3) diagnostics for pinning down artifactual point configurations, and 4) restricting MDS to subsets of objects and subsets of dissimilarities.

MDS was originally developed for the social sciences, but it is now also used for laying out graphs. Graph layout is usually done in 2-D, but we allow layouts in arbitrary dimensions. We show applications to the mapping of computer usage data, to the dimension reduction of marketing segmentation data, to the layout of mathematical graphs and social networks, and finally to the spatial reconstruction of molecules.

XGvis uses the XGobi system for visualizing point configurations. The systems run under Linux and Unix operating systems and they are freely available from the following site:
http://www.research.att.com/areas/stat/xgobi/

Key Words: Multidimensional Unfolding, Proximity Data, Multivariate Analysis, Interactive Data Visualization, Dimension Reduction, Graph Layout, Social Networks, Molecular Conformation.

---

[1]Andreas Buja is Technology Consultant, AT&T Labs, 180 Park Ave., P.O. Box 971, Florham Park, NJ 07932-0971. (andreas@research.att.com, http://www.research.att.com/~andreas)

[2]Deborah F. Swayne is Senior Technical Staff Member, AT&T Labs, 180 Park Ave., P.O. Box 971, Florham Park, NJ 07932-0971. (dfs@research.att.com, http://www.research.att.com/~dfs)

[3]Michael L. Littman is Principal Technical Staff Member, AT&T Labs, 180 Park Ave., P.O. Box 971, Florham Park, NJ 07932-0971. (mlittman@research.att.com, http://www.research.att.com/~mlittman)

[4]Nathaniel Dean is Associate Professor, Computational and Applied Mathematics - MS 134, Rice University, 6100 Main Street, Houston, TX 77005. (nated@caam.rice.edu, http://www.caam.rice.edu/~nated)

[5]Heike Hofmann is Postdoc, AT&T Labs, 180 Park Ave., P.O. Box 971, Florham Park, NJ 07932-0971. (heike@research.att.com, http://www1.math.uni-augsburg.de/~hofmann)

1 Introduction: Basics of Multidimensional Scaling

The present section gives a short introduction to those types of MDS that are relevant for this article. Section 2 gives an overview of how a user operates the XGvis system. Section 3 deals with algorithm animation, direct manipulation and perturbation of the configuration. Section 4 gives details about the cost functions and their interactively controlled parameters for transformation, subsetting and weighting of dissimilarities. Section 5 describes diagnostics for MDS. Section 6 is about computational and systems aspects, including coordination of windows, algorithms, and large data problems. Finally, Section 7 gives a tour of applications with examples of proximity analysis, dimension reduction, and graph layout in two and more dimensions.

1.1 Proximity Data and Stress Functions

Multidimensional scaling (MDS) is a family of methods for analyzing proximity data. Proximity data consist of similarity or, equivalently, dissimilarity information for *pairs of objects.* This contrasts with multivariate data that consist of covariate information for *individual objects.* If the objects are labeled $i = 1, ..., N$, proximity data can be assumed to be dissimilarity values $D_{i,j}$. If the data are given as similarities, some monotone decreasing transformation will convert them to dissimilarities. Dissimilarity data occur in many areas (see Section 1.3).

The goal of MDS is to map the objects $i = 1, ..., N$ to points $\mathbf{x}_1, ..., \mathbf{x}_N \in \mathbb{R}^k$ in such a way that the given dissimilarities $D_{i,j}$ are well-approximated by the distances $\|\mathbf{x}_i - \mathbf{x}_j\|$. Psychometricians often call these distances the "model" fitted to the data $D_{i,j}$.

The dissimilarity matrices of Figure 1 are simple examples with easily recognized error-free MDS solutions: the left matrix suggests mapping the five objects to an equispaced linear arrangement; the right matrix suggests mapping the three objects to a right triangle. The figure shows configurations actually found by MDS. The first configuration can be embedded in $k = 1$ dimension, while the second needs at least $k = 2$ dimensions. The choice of embedding dimension k is arbitrary in principle, but low in practice: $k = 1, 2, 3$ are the most frequently used dimensions, for the simple reason that the points serve as easily visualized representors of the objects.

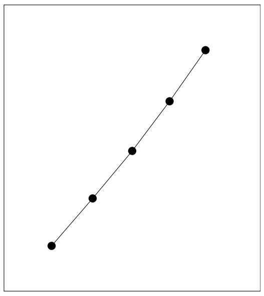
In real data, there are typically many more objects, and the dissimilarities usually contain error as well as bias with regard to the fitted distances.

The oldest version of MDS, called classical scaling, is due to Torgerson (1952). It is, however, a later version due to Kruskal (1964a,b) that has become the leading MDS method. It is defined in terms of minimization of a cost function called "Stress", which is simply a measure of lack of fit between dissimilarities $D_{i,j}$ and distances $\|\mathbf{x}_i - \mathbf{x}_j\|$. In the simplest case, Stress is a residual sum of squares:

$$\mathrm{Stress}_D(\mathbf{x}_1, ..., \mathbf{x}_N) \;=\; \left(\sum_{i \neq j = 1..N} \left(D_{i,j} - \|\mathbf{x}_i - \mathbf{x}_j\|\right)^2 \right)^{1/2} \qquad (1)$$

where the outer square root is just a convenience that gives greater spread to small values. For a given dissimilarity matrix $D = (D_{i,j})$, MDS minimizes Stress over all point configurations $(\mathbf{x}_1, ..., \mathbf{x}_N)$, thought of as $k \times N$-dimensional hypervectors of unknown parameters. The

$$D = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 0 & 1 & 2 & 3 \\ 2 & 1 & 0 & 1 & 2 \\ 3 & 2 & 1 & 0 & 1 \\ 4 & 3 & 2 & 1 & 0 \end{bmatrix} \qquad D = \begin{bmatrix} 0 & 3 & 4 \\ 3 & 0 & 5 \\ 4 & 5 & 0 \end{bmatrix}$$



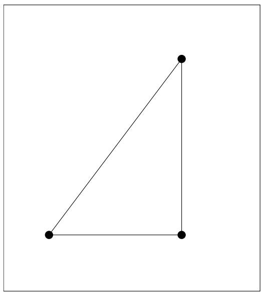


Figure 1: *Simple Examples of Dissimilarity Matrices and Their Optimal Scaling Solutions.*

minimization can be carried out by straightforward gradient descent applied to Stress_D, viewed as a function on $\mathbb{R}^{kN}$.

We note a technical detail: MDS is blind to asymmetries in the dissimilarity data because

$$(D_{i,j} - \|\mathbf{x}_i - \mathbf{x}_j\|)^2 + (D_{j,i} - \|\mathbf{x}_j - \mathbf{x}_i\|)^2 \;=\; 2 \cdot ((D_{i,j} + D_{j,i})/2 - \|\mathbf{x}_i - \mathbf{x}_j\|)^2 + \ldots,$$

where $\ldots$ is an expression that does not depend on $\|\mathbf{x}_i - \mathbf{x}_j\|$. Without loss of generality we assume from now on that the dissimilarities are symmetric: $D_{i,j} = D_{j,i}$. If they are not, they should be symmetrized by forming pairwise averages. The assumption of symmetry will later be broken in one special case, when one of the two values is permitted to be missing (Section 4.4).

1.2 Types of Multidimensional Scaling

There exist many MDS methods, differing mostly in the cost function they use. Here are two dichotomies that allow us to structure some possibilities:

- **Kruskal-Shepard distance scaling** versus **classical Torgerson-Gower inner-product scaling**: Distance scaling is based on direct fitting of distances to dissimilarities, whereas the older classical scaling is based on a detour whereby dissimilarities are converted to a form that is naturally fitted by inner products $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ (see below).
- **Metric scaling** versus **nonmetric scaling**: Metric scaling uses the actual values of the dissimilarities, while nonmetric scaling effectively uses only their ranks (Shepard 1962, Kruskal 1964a). Nonmetric MDS is realized by estimating an optimal monotone transformation $f(D_{i,j})$ of the proximities simultaneously with the configuration.

In XGvis we implemented both distance scaling and classical inner-product scaling, and both metric and nonmetric scaling. In all, four types of MDS are provided: metric distance, nonmetric distance, metric classical, and nonmetric classical scaling. The unusual case of nonmetric classical scaling will be described in Section 4.2.

A *conceptual difference* between classical and distance scaling is that inner products rely on an origin, while distances do not; a set of inner products determines uniquely a set of distances, but a set of distances determines a set of inner products only, modulo change of origin. To avoid arbitrariness, one constrains classical scaling to configurations with the mean at the origin.

The *computational difference* between classical and distance scaling is that the minimization problem for classical scaling can be solved with a simple eigendecomposition, while distance scaling requires iterative minimization. In XGvis, though, classical scaling is implemented with iterative gradient descent of a cost function called "Strain", as is distance scaling with regard to Stress. This computational uniformity has advantages: it is straightforward to introduce weights and missing values in Strain and Stress, but not so in eigendecompositions. Both weights and missing values are extensively used in XGvis.

1.3 Applications of MDS

Here is an incomplete list of application areas for MDS:

- MDS was invented for the *analysis of proximity data* which arise in the following areas:
  - *The social sciences:* Proximity data take the form of similarity ratings for pairs of stimuli such as tastes, colors, sounds, people, nations, ...
  - *Archaeology:* Similarity of two digging sites can be quantified based on the frequency of shared features in artifacts found in the sites.
  - *Classification problems:* In classification with large numbers of classes, pairwise misclassification rates produce confusion matrices that can be analyzed as similarity data. An example would be confusion rates of phonemes in speech recognition.
- Another early use of MDS was for *dimension reduction*: Given high-dimensional data $\mathbf{y}_1, ..., \mathbf{y}_N \in \mathbb{R}^K$ (K large), compute a matrix of pairwise distances $\mathrm{dist}(\mathbf{y}_i, \mathbf{y}_j) = D_{i,j}$, and use distance scaling to find lower-dimensional $\mathbf{x}_1, ..., \mathbf{x}_N \in \mathbb{R}^k$ ($k << K$) whose pairwise distances reflect the high-dimensional distances $D_{i,j}$ as well as possible. In this application, distance scaling is a non-linear competitor of principal components. Classical scaling, on the other hand, is identical to principal components when used for dimension reduction. For a development of multivariate analysis from the point of view of distance approximation, see Meulman (1992).
- In chemistry, MDS can be used for *molecular conformation*, that is, the problem of reconstructing spatial structure of molecules. This situation differs from the above areas in that 1) actual distance information is available from experiments or theory, and 2) the only meaningful embedding dimension is $k = 3$, physical space. Configurations are here called "conformations." Some references are Crippen and Havel (1978), Havel (1991), Glunt et al. (1993), and Trosset (1998a). See also our example in Section 7.4.

- A fourth use of MDS is for *graph layout*, an active area at the intersection of discrete mathematics and network visualization, see Di Battista et al. (1994). An early example before its time was Kruskal and Seery (1980). From graphs one can derive distances, such as shortest-path metrics, which can be subjected to MDS for planar or spatial layout. Note that shortest-path metrics are generally strongly non-Euclidean, hence significant residual should be expected in this type of application.

In Section 7 we will show examples of data in all four categories. Our overall experience with the various types of MDS is as follows: distance scaling is not particularly good at dimension reduction, whereas classical scaling is not particularly good at graph layout. The examples will exemplify these points.

1.4 A First Example: The Rothkopf Morse Code Data

For illustration we use the inescapable Rothkopf Morse code data (1957) as our running example. They are available with the XGobi/XGvis distribution from the site mentioned in the abstract. These data are to MDS what Fisher's Iris data are to discriminant analysis. The data originated in an experiment where inexperienced subjects were exposed to pairs of Morse codes in rapid order; the subjects had to decide whether the two codes in a pair were identical or not. The resulting data were summarized in a table of confusion rates.

Confusion rates are similarity measures: codes that are often confused are interpreted as "similar" or "close." Similarities need to be converted to dissimilarities. In principle any monotone decreasing transformation can be used for conversion, but we used the following:

$$D_{i,j}^2 = s_{i,i} + s_{j,j} - 2s_{i,j} \ .$$

This yielded all non-negative values because the confusion matrix $(s_{i,j})_{i,j}$ is diagonally dominant (most identical code pairs are correctly judged even by inexperienced subjects). Unlike other conversion methods, this one has the desirable property $D_{i,i} = 0$. We also symmetrized the dissimilarities before the conversion as MDS responds only to the symmetric part of proximity data.

Applying all four scaling methods in $k = 2$ dimensions to the Morse code dissimilarities produced the configurations shown in Figure 2. We decorated the plots with labels and lines to aid interpretation. In particular, we connected groups of codes of the same length, except for codes of length four which we broke up into three groups and a singleton. We observe that, roughly, the length of the codes increases left to right, and the fraction of dots increases bottom up. Both of these observations agree with the many published accounts (Shepard 1962, Kruskal and Wish 1978, p. 13, Borg and Lingos 1987, p. 69, Borg and Groenen 1997, p. 59, for example). The orientation of the configurations in $\mathbb{R}^2$ is arbitrary due to rotation invariance of the distances $\|\mathbf{x}_i - \mathbf{x}_j\|$ and inner products $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$. We were therefore free to rotate the configurations in order to achieve this particular interpretation of the horizontal and vertical axes.

We note that the configurations produced by the four scaling methods show significant differences. Nonmetric distance scaling (top right) produces probably the most satisfying configuration, with the exception of the placement of the codes of length 1 ("E" and "T").

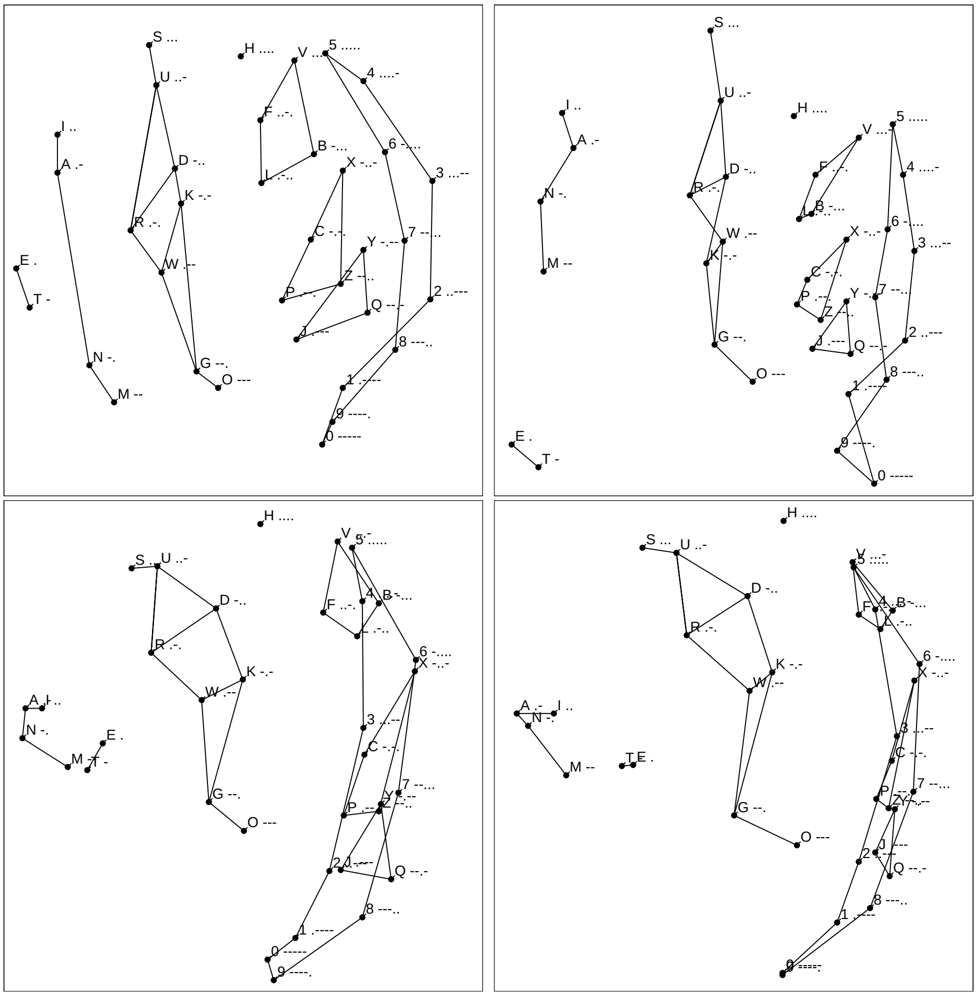


Figure 2: *Rothkopf's Morse Code Data: four 2-D configurations produced with XGvis. Top row: metric and nonmetric distance scaling. Bottom row: metric and nonmetric classical scaling.*

Metric scaling (top left) suffers from circular bending but it places "E" and "T" in the most plausible location. The classical scaling methods (bottom row) bend the configurations in different ways. More problematically, they overlay the codes of length 4 and 5 and invert the placement of the codes of length 1 and 2, both of which seem artifactual. In fact they aren't: classical scaling requires a third dimension to distinguish between these two pairs of groups. Distance scaling is better at achieving compromises in lower dimensions, whereas classical scaling is more rigid in this regard.

2 Operation of XGvis

The main use of MDS configurations is for visualization. Because configurations are essentially multivariate data, any MDS system calls for a multivariate data visualization tool. Two of us being co-authors of the XGobi system for data visualization, it was natural that

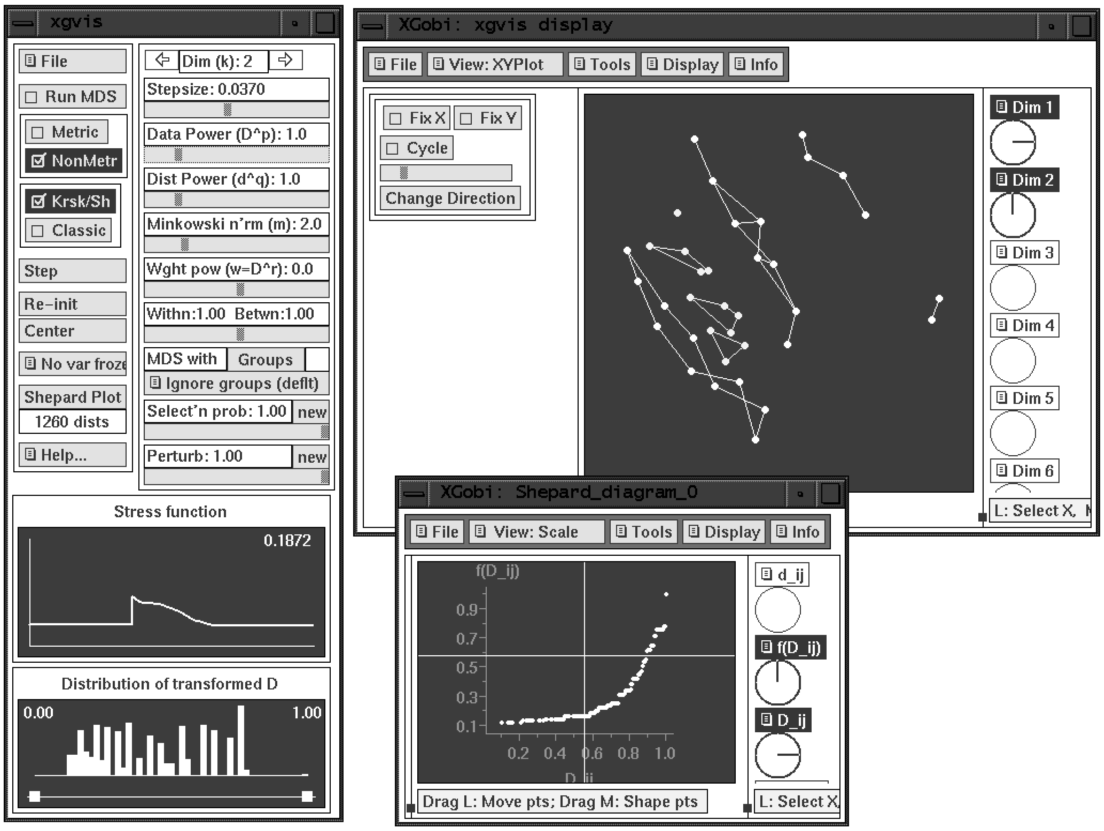


Figure 3: *The Major XGvis Windows. On the left is the master control panel, on the right is the XGobi window for the configuration. Below is the XGobi window for an optional Shepard plot.*

we chose XGobi as the viewing engine underlying XGvis (Swayne, Cook and Buja 1998, Buja, Cook and Swayne 1996). We thus conceived of XGvis as a master program that creates and feeds XGobi windows. Figure 3 shows how this presents itself to the user: a master panel with MDS controls (left), an XGobi window for viewing the configuration (right), and a discretionary XGobi window for diagnostics (bottom).

In what follows, we make XGvis and XGobi operations recognizable by placing them in quotation marks. The basic sequence of MDS interactions is as follows:

- Start up with dissimilarity data, multivariate data or graph data. Correspondingly, XGvis will perform proximity analysis, dimension reduction, or graph layout. Provide an initial configuration, or else a random configuration is generated automatically.
- Select one of the four scaling methods. The default is metric distance scaling.
- Choose a dimension. The default is 3.
- Initiate optimization ("Run MDS") and watch the animation of the configuration and the progression of the Stress or Strain value. When the shape of the configuration stops changing, slow the optimization down by lowering the stepsize interactively. Fi-

nally, stop the optimization (toggle "Run MDS"). XGvis does not have an automatic convergence criterion, and optimization does not stop on its own.

- Examine the shape of the optimized configuration: If the chosen dimension k is two, remain in "XY Plot". If k is three, use 3-D "Rotation". If k is higher, use the "Grand Tour".
- Interpret the configuration: Assuming informative object labels were provided with the input data, search the configuration by labeling points ("Identify"). If covariates are available in addition to the dissimilarities, interpretation can be further aided by linked color brushing between covariate views and configuration views ("Brush"). Multiple XGobi windows are automatically linked for brushing and labeling. As this is only a tentative search for interpretable structure, use "transient" brushing.
- Enhance the configuration: After acquiring a degree of familiarity with the configuration, use "persistent" brushing in order to permanently characterize subsets of interest. Enhance the configuration further by persistently labeling interesting points. Finally, enhance the overall perception of shape by connecting selected pairs of nearby points with lines ("Line Editing") and coloring the lines ("Brush").
- Turn on optimization and leave it continually running. Observe the effects of
  - experimenting with various parameters,
  - subsetting objects,
  - subsetting dissimilarities,
  - weighting dissimilarities,
  - manually moving points and groups of points.
  - perturbing the configuration or restarting from random configurations.
- Stop optimization and perform diagnostics by generating a separate XGobi window that shows among other things a "Shepard plot" of the transformed dissimilarities and the fitted distances.

We described elsewhere (Swayne et al. 1998) XGobi operations such as three-D rotations and grand tours, as well as (linked) brushing, labeling and line editing. Moving points is also an XGobi operation, but in conjunction with MDS optimization it takes on a special importance and is therefore described in Section 3. MDS parameters as well as weighting and subsetting of dissimilarities affect the cost function and are therefore specific to MDS. They are the subject of Section 4.

3 Animated Optimization and Point Manipulation

At the time of writing, most software for MDS still works with batch processing, even though many older programs have been wrapped in somewhat interactive PC environments. A pioneering system that is truly interactive in our meaning of the term is McFarlane and Young's "ViSta-MDS" (1994). Development of this system took place around the time when

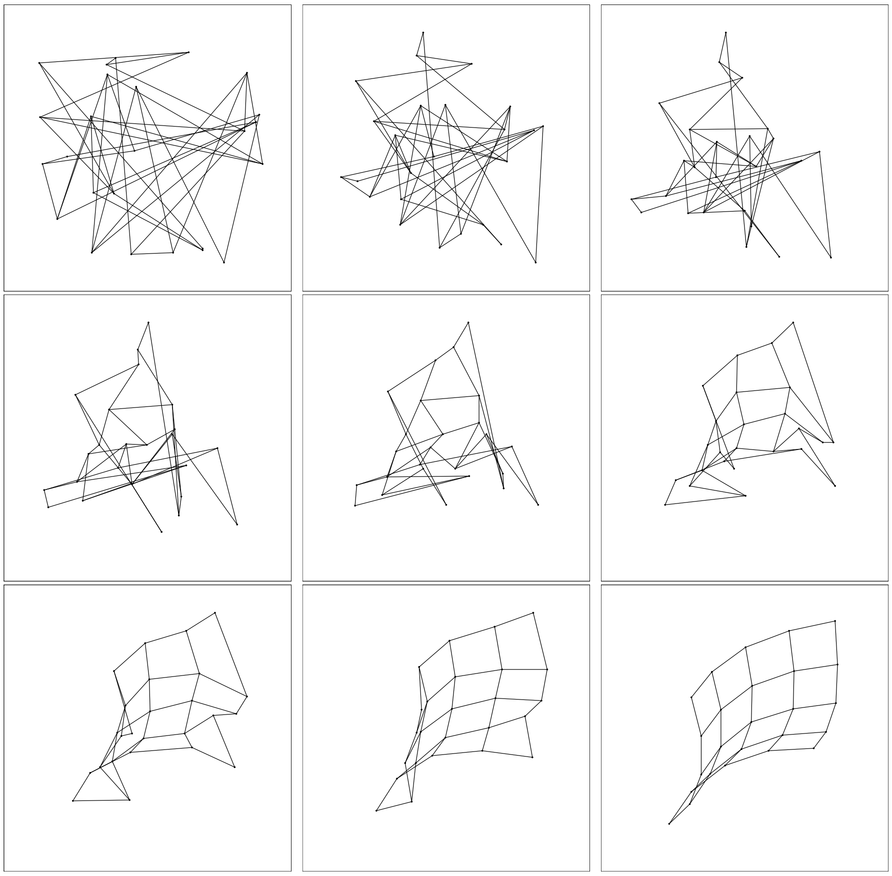


Figure 4: *Snapshots from a MDS Animation. The figure shows nine stages of a Stress minimization in three dimensions. It reconstructed a* 5×5 *square grid from a random configuration. The grid was defined as a graph with 25 nodes and 40 edges. The distances were computed as the lengths of the minimal paths in the graph. These distances are not Euclidean, causing curvature in the configuration.*

we developed a first version of XGvis (Littman et al. 1992). Although the two systems were developed independently, they share two important interactive capabilities:

1. **Animated Optimization:** The configuration points are displayed continuously as they are subjected to MDS optimization. For a series of stills from an animated optimization, see Figure 4. At the same time, the values of the cost function are also shown in a trace plot (Figure 3).

2. **Manual Dragging:** Configuration Points can be moved interactively with mouse dragging.

McFarlane and Young call this methodology "sensitivity analysis" because moving points and observing the response of the optimization amounts to checking the stability of the configuration.

ViSta-MDS implements a *two-step mode of operation* in which users alternate between animated optimization and manipulation. XGvis permits the two-step mode also, but in

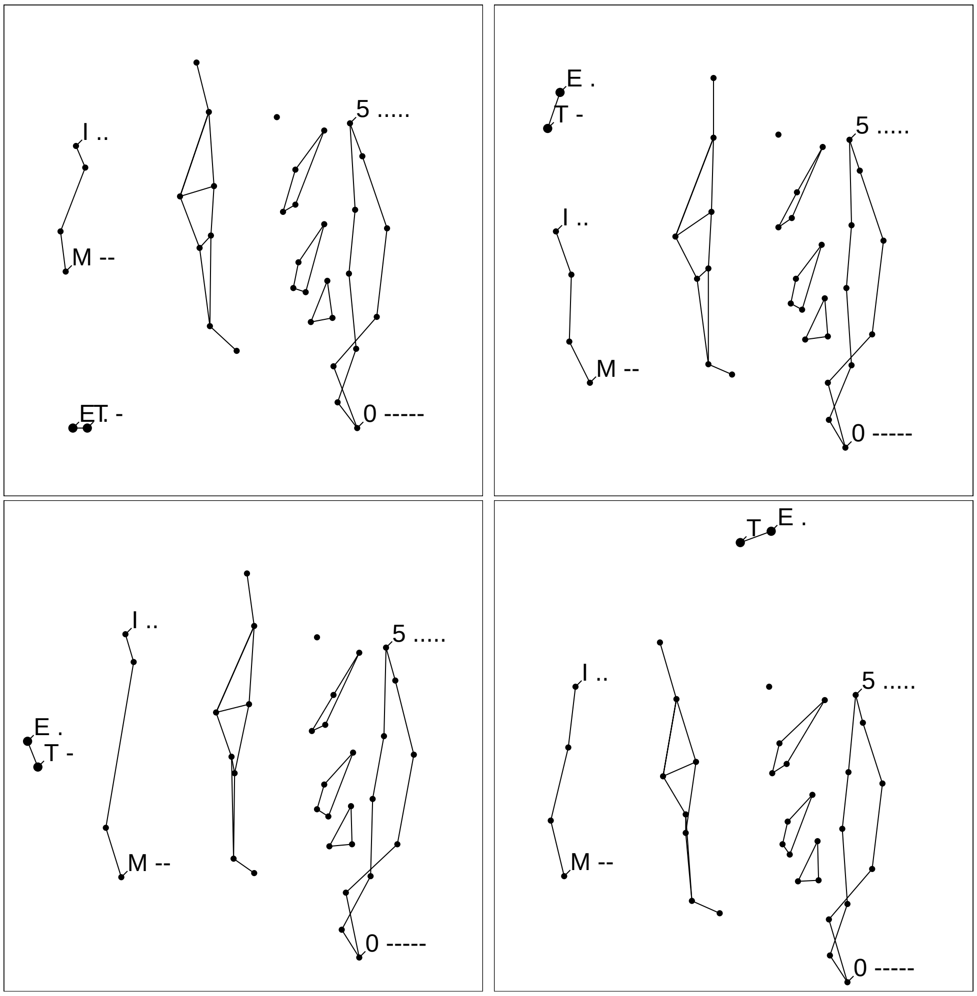


Figure 5: *Four Local Minima Found by Moving the Group $\{E,T\}$ into Different Locations. The Stress values are, left to right and top to bottom: 0.2101, 0.2187, 0.2189, 0.2207.*

addition it implements a *fluid mode of operation* in which the user runs a never-ending optimization loop, with no stopping criterion whatsoever. The user manipulates the configuration points while the optimization is in progress. The optimizer ignores the manipulated point, but the other points "feel" the dislocation through the change in the cost function, and they are therefore slowly dragged. They try to position themselves in a local minimum configuration with regard to the manipulated point. As soon as the manipulated point is let go, it snaps into a position that turns the configuration into a local minimum of the cost function. The resulting feel for the user is that of pricking and tearing the fabric of the configuration while exploring deformations of its shape.

In addition to moving one point at a time, XGvis permits moving groups of points with shared glyphs and colors. This capability can be helpful when it has become evident that certain points always stay together, as in the Morse code data the points for the codes of length one ("E" and "T"). The only way to find further local minima is by moving the points jointly. Indeed, when the Morse code data are mapped into two dimensions with metric distance scaling and a third power transformation (which mimics a nonmetric solution very well), there exist four different locally optimal locations for the codes of length one (Figure 5). Another two local minimum configurations can be found by moving the codes of length two.

More local minima can be found by restarting optimization with a random configuration. XGvis provides Gaussian random point clouds (with a mouse click on "new", next to "Perturb", Figure 3). In addition, it is possible to examine local stability by perturbing a configuration with normal random numbers by forming a convex mixture of the present configuration and a random Gaussian configuration. The mixing fraction is read from a slider (below "Perturb"). The default mixing parameter is 100% random, which means a completely new random configuration is generated. A smaller fraction of 20% or so can be used for local stability checks: if optimization always drives the perturbed configuration back to its previous state, it is stable under 20% perturbation. Further stability checks will be discussed below. For a discussion of the problem of local minimum configurations, see Borg and Groenen (1997), Section 13.4.

As a final remark we point out another use of manual dragging: rotation of configurations for interpretability, essentially the factor rotation problem translated to MDS. After examining a configuration and decorating it with labels, colors, glyphs and lines, one usually obtains interpretations of certain directions in configuration space. This is when one develops a desire to rotate the configuration to line these directions up with the horizontal and vertical axes. One can achieve this by dragging points while optimization is running continuously. When points are moved gently, the optimization will try to translate and rotate the configuration so that the moved point maintains a local minimum location relative to the other points. We used this effect to rotate the four configurations of Figure 2 into their present orientations.

4 Cost Functions: Stress and Strain

As mentioned in the introduction, we use iterative minimization of cost functions even where eigendecompositions would work, the reason being that missing and weighted dissimilarities are handled with difficulty by the latter but trivially by the former. We develop the specific forms of Stress and Strain used in XGvis.

4.1 Stress

Although the simplest from of cost function for distance scaling is a residual sum of squares, it is customary to report Stress values that are standardized and unit-free. This may take the form

$$\mathrm{Stress}_D(\mathbf{x}_1, ..., \mathbf{x}_N) \;=\; \left(\frac{\sum_{i,j} (D_{i,j} - \|\mathbf{x}_i - \mathbf{x}_j\|)^2}{\sum_{i,j} D_{i,j}^2} \right)^{1/2}$$

In this form of Stress one can explicitly optimize the size of the configuration:

$$\min_t \; \mathrm{Stress}_D(t \cdot \mathbf{x}_1, ..., t \cdot \mathbf{x}_N) \;=\; \left(1 \;-\; \frac{\left(\sum_{i,j} D_{i,j} \cdot \|\mathbf{x}_i - \mathbf{x}_j\|\right)^2}{\sum_{i,j} D_{i,j}^2 \cdot \sum_{i,j} \|\mathbf{x}_i - \mathbf{x}_j\|^2} \right)^{1/2}$$

This can be interpreted geometrically as the sine between the vectors $\{D_{i,j}\}_{i,j}$ and $\{\|\mathbf{x}_i - \mathbf{x}_j\|\}_{i,j}$, which is hence a number between zero and one. It is the form of Stress reported and traced by XGvis.

The Stress actually used in XGvis is of considerably greater generality, however: it permits 1) power transformations of the dissimilarities in metric mode and isotonic transformations in nonmetric mode, 2) Minkowski distances in configuration space, 3) powers of the distances, 4) weighting of the dissimilarities, and 5) missing and omitted dissimilarities. We give the complete formula for Stress as implemented in XGvis and defer the explanation of details to Section 4.3:

$$\mathrm{STRESS}_D(\mathbf{x}_1, ..., \mathbf{x}_N) \; = \; \left(1 - \cos^2\right)^{1/2}$$

$$\cos^2 \; = \; \frac{\left(\sum_{(i,j)\in I} \; w_{i,j} \cdot f(D_{i,j}) \cdot \|\mathbf{x}_i - \mathbf{x}_j\|_m^q \right)^2}{\left(\sum_{(i,j)\in I} \; w_{i,j} \cdot f(D_{i,j})^2\right) \left(\sum_{(i,j)\in I} \; w_{i,j} \cdot \|\mathbf{x}_i - \mathbf{x}_j\|_m^{2q}\right)}$$

$D_{i,j} \in \mathbb{R}, \;\; \geq 0, \;\; N \times N$ matrix of dissimilarity data

$$f(D_{i,j}) \; = \; \begin{cases} D_{i,j}^p \;, & \text{for metric MDS} \\ s \cdot \mathrm{Isotonic}(D_{i,j}) + (1-s) \cdot D_{i,j}^p \;, & \text{for nonmetric MDS} \end{cases}$$

$0 \leq p \leq 6$, default: $p = 1$ (no transformation)
$0 \leq s \leq 1$, default: $s = 1$ (fully isotonic transformation)
Isotonic = monotone $\uparrow$ transformation estimated with isotonic regression

$\mathbf{x}_1, ..., \mathbf{x}_N \in \mathbb{R}^k$, configuration points; $\;1 \leq k \leq 12$, default: $k = 3$

$$\|\mathbf{x}_i - \mathbf{x}_j\|_m^q \; = \; \Big(\sum_{\nu=1,...,k} |x_{i,\nu} - x_{j,\nu}|^m\Big)^{q/m}, \qquad \text{configuration distances, } (..)^q$$

$1 \leq m \leq 6$, $\;\; m = 2$: Euclidean (default)
$\qquad\qquad\quad m = 1$: City block
$0 \leq q \leq 6$, $\;\; q = 1$: common Stress (default)
$\qquad\qquad\quad q = 2$: so-called SStress

The summation set I and the weights $w_{i,j}$ will be discussed in Section 4.5.

4.2 Strain, Metric and Nonmetric

Classical scaling is based on inner products, which unlike distances depend on the origin. One standardizes the procedure by assuming configurations with zero mean: $\sum_i \mathbf{x}_i = 0$. Under this assumption, one fits inner products $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ to a transformation of the dissimilarity data. The derivation of this transformation is based on the following heuristic:

$$D_{i,j}^2 \;\approx\; \|\mathbf{x}_i - \mathbf{x}_j\|^2 = \|\mathbf{x}_i\|^2 + \|\mathbf{x}_j\|^2 - 2\langle \mathbf{x}_i, \mathbf{x}_j \rangle$$

As the matrix $(\langle \mathbf{x}_i, \mathbf{x}_j \rangle)_{i,j}$ has zero row and column means, one recognizes that the necessary transformation is simply the removal of row and column means from the matrix

$$\tilde{D}_{i,j} := -D_{ij}^2/2 \ ,$$

which is commonly known as "double-centering":

$$B_{ij} = \tilde{D}_{ij} - \tilde{D}_{i\bullet} - \tilde{D}_{\bullet j} + \tilde{D}_{\bullet\bullet} \ ,$$

where $\tilde{D}_{i\bullet}$, $\tilde{D}_{\bullet j}$ and $\tilde{D}_{\bullet\bullet}$ are row, column and grand means. The quantities $B_{i,j}$ are now plausibly called "inner-product data":

$$B_{i,j} \ \approx \ \langle \mathbf{x}_i, \mathbf{x}_j \rangle$$

We call "Strain" any cost function that involves inner products of configuration points and inner-product data. One form of Strain is a standardized residual sum of squares between the quantities $B_{i,j}$ and the inner products:

$$\text{Strain}_D(\mathbf{x}_1, ..., \mathbf{x}_N) \ = \ \left(\frac{\sum_{i,j} (\, B_{i,j} - \langle \mathbf{x}_i, \mathbf{x}_j \rangle)^2}{\sum_{i,j} \langle \mathbf{x}_i, \mathbf{x}_j \rangle^2} \right)^{1/2}$$

In size-minimized form this is:

$$\min_t \ \text{Strain}_D(t \cdot \mathbf{x}_1, ..., t \cdot \mathbf{x}_N) \ = \ \left(1 \ - \ \frac{\left(\sum_{i,j} B_{i,j} \cdot \langle \mathbf{x}_i, \mathbf{x}_j \rangle \right)^2}{\sum_{i,j} B_{i,j}^2 \cdot \sum_{i,j} \langle \mathbf{x}_i, \mathbf{x}_j \rangle^2} \right)^{1/2}$$

This is not the form of Strain we use, however. Instead, we pose ourselves the problem of finding a form that can be made nonmetric. Nonmetric classical MDS seems a contradiction in terms because classical scaling must be metric according to the conventional interpretations. Yet we may ask whether it would not be possible to transform the dissimilarity data $D_{i,j}$ in such a way that a better Strain could be obtained, just as the data are transformed in nonmetric distance scaling to obtain a better Stress. This is indeed possible, and a solution has been given by Trosset (1998b). We give here an alternative derivation with additional simplifications that permit us to fit the Strain minimization problem in the existing framework. Classical nonmetric scaling is not a great practical advance, but it fills a conceptual gap. It also gives the software a more satisfactory structure by permitting all possible pairings of {metric, nonmetric} with {classical scaling, distance scaling}. The properties of nonmetric classical scaling are not well-understood at this point. The implementation in XGvis will hopefully remedy the situation.

We start with the following observations:

- Because the matrix $(B_{i,j})_{i,j}$ is doubly-centered, any Strain-minimizing configuration is centered at the origin.
- If, however, the configuration is constrained to be centered at the origin, one may use $(-D_{i,j}^2/2)_{i,j}$ instead of $(B_{i,j})_{i,j}$ in the Strain formula; the minimizing configurations will be the same. That is, double centering of $(-D_{i,j}^2/2)_{i,j}$ may be replaced with forced centering of the configuration.

The first bullet is almost tautological in view of our introduction of classical scaling. The second bullet may be new; it is certainly the critical one. To elaborate we need a little linear algebra:

If A and C are $N \times N$ matrices, denote by $\langle A, C\rangle_F = \mathrm{trace}(A^T C) = \sum_{i,j} A_{i,j} C_{i,j}$ the Frobenius inner product, and by $\|A\|_F = \langle A, A\rangle^{1/2}$ the Frobenius norm. Furthermore, let $\mathbf{e} = (1, ..., 1)^T \in \mathrm{I\!R}^N$ and I_N be the $N \times N$ identity matrix, so that $P = I_N - \mathbf{e}\mathbf{e}^T/N^{1/2}$ is the centering projection. Then the equation $B_{i,j} = \tilde{D}_{i,j} - \tilde{D}_{i\bullet} - \tilde{D}_{\bullet j} + \tilde{D}_{\bullet\bullet}$ can be re-expressed as $B = P\tilde{D}P$. Finally, let X be the $N \times k$ configuration matrix whose rows contain the coordinates of the configuration points, so that $(\langle \mathbf{x}_i, \mathbf{x}_j\rangle)_{i,j} = XX^T$. The centering condition $\sum_i \mathbf{x}_i = 0$ can be re-expressed as $PX = X$, and the residual sum of squares of Strain as

$$\sum_{i,j} (\, B_{i,j} - \langle \mathbf{x}_i, \mathbf{x}_j\rangle)^2 \;=\; \|B - XX^T\|_F^2 \;.$$

Using repeatedly a basic property of traces, $\mathrm{trace}(A^T C) = \mathrm{trace}(C^T A)$, one derives $\langle B, XX^T\rangle_F = \langle B, PXX^T P\rangle_F$ and hence:

$$\|B - XX^T\|_F^2 = \|B - PXX^T P\|_F^2 \;+\; \|PXX^T P - XX^T\|_F^2 \;.$$

Therefore, a minimizing configuration matrix satisfies $PX = X$, which shows the first bullet above. For the second bullet, assume $PX = X$ and observe that $\langle \tilde{D}, XX^T\rangle_F = \langle P\tilde{D}P, XX^T\rangle_F = \langle B, XX^T\rangle_F$. It follows

$$\|\tilde{D} - XX^T\|_F^2 \;=\; \|\tilde{D} - B\|_F^2 \;+\; \|B - XX^T\|_F^2 \;.$$

Therefore, under the constraint $\sum_i \mathbf{x}_i = 0$ minimizing $\sum_{i,j}((-D_{i,j}^2/2) - \langle \mathbf{x}_i, \mathbf{x}_j\rangle)^2$ is the same as minimizing $\sum_{i,j}(B_{i,j} - \langle \mathbf{x}_i, \mathbf{x}_j\rangle)^2$ as the two cost functions differ only by a constant.

The former cost function has the property we were looking for: it lends itself to a non-metric extension by replacing the original transformation $-D_{i,j}^2/2$ with a general descending transformation $f(-D_{i,j})$, where f is a monotone increasing (non-decreasing) function that can be estimated with isotonic regression of $\langle \mathbf{x}_i, \mathbf{x}_j\rangle$ on $-D_{i,j}$. In the metric case, an extension to power transformations is natural: $f(-D_{i,j}) = -D_{i,j}^{2p}$. Thus we have solved the problem of finding a natural nonmetric form of classical scaling.

Like Stress, the Strain actually implemented in XGvis is a standardized size-optimized version (to avoid the collapse of $f(-D_{i,j})$), and it also has some generalizations such as weighting and omitting of dissimilarities:

$$\mathrm{STRAIN}_D(\mathbf{x}_1, ..., \mathbf{x}_N) \;=\; \left(1 - \cos^2\right)^{1/2}$$

$$\cos^2 \;=\; \frac{\left(\, \sum_{(i,j)\in I} \; w_{i,j} \cdot f(-D_{i,j}) \cdot \langle \mathbf{x}_i, \mathbf{x}_j\rangle \,\right)^2}{\left(\sum_{(i,j)\in I} \; w_{i,j} \cdot f(-D_{i,j})^2\right)\left(\sum_{(i,j)\in I} \; w_{i,j} \cdot \langle \mathbf{x}_i, \mathbf{x}_j\rangle^2\right)}$$

$D_{i,j} \in \mathrm{I\!R}, \;\; \geq 0, \;\; N \times N$ matrix of dissimilarity data

$$
f(-D_{i,j}) \;=\; \begin{cases} -D_{i,j}^{2p} \,, & \text{for metric MDS} \\ s \cdot \text{Isotonic}(-D_{i,j}) + (1-s)\cdot(-D_{i,j}^{2p}) \,, & \text{for nonmetric MDS} \end{cases}
$$

$$
\begin{aligned}
& 0 \le p \le 6,\ \text{default: } p = 1 \text{ (no transformation)} \\
& 0 \le s \le 1,\ \text{default: } s = 1 \text{ (fully isotonic transformation)} \\
& \text{Isotonic} = \text{monotone} \uparrow \text{transformation estimated} \\
& \qquad\qquad\quad \text{with isotonic regression}
\end{aligned}
$$

$$
\mathbf{x}_1, ..., \mathbf{x}_N \in \mathbb{R}^k,\ \text{configuration points, constrained to } \sum \mathbf{x}_i = 0
$$

$$
1 \le k \le 12,\ \text{default: } k = 3
$$

$$
\langle \mathbf{x}_i, \mathbf{x}_j \rangle \;=\; \sum_{\nu=1,...,k} x_{i,\nu} \cdot x_{j,\nu} \,, \qquad \text{configuration inner products}
$$

The summation set I and the weights $w_{i,j}$ will be discussed in Section 4.5.

4.3 Parameters of Stress and Strain

In XGvis both Stress and Strain have parameters that are under interactive user control. See Figure 3 for the layout of buttons and sliders for parameter choices.

- The most fundamental "parameters" are the discrete choices of metric versus nonmetric and distance versus classical scaling. The default is metric distance scaling.

- Next in importance is the choice of the dimension, k. The convention is $k = 2$, but because of the availability of 3-D rotations in XGobi we chose $k = 3$ as the default. For $k = 2$ the user should choose "X-Y plots", and for $k \ge 4$ the so-called "grand tour", a generalization of 3-D rotations to higher dimensions. The maximum for k is 12, but it takes just a few seconds to recompile the code with another maximum dimension.

- Both metric and nonmetric scaling are each implemented with control of a parameter that affects the transformation of the dissimilarities. The two parameters are very different in nature; just the same, they are read from the same slider which changes its meaning depending on whether metric or nonmetric scaling is chosen. Confusions are not possible.

  - *Metric scaling* is implemented with a *power transformation* with exponent p. The value $p = 1$ (default) corresponds to no transformation. The range of p is limited to the interval $[0, 6]$. In our experience powers as high as 4 have proven useful, which is why we chose an interval just slightly larger. A very interesting power is 0: It describes objects that form a simplex, that is, every object is equally far from every other object. This is the "null case" of total indiscrimination.
  - *Nonmetric scaling* is implemented with a parameter s for *mixing of the isotonic transformation* $f(D_{i,j})$ *with the metric power transform* $D_{i,j}^p$. The value $s = 1$ (default) corresponds to the isotonic transformation (purely nonmetric scaling), and $s = 0$ to the pure power transformation (purely metric scaling). Sliding s

across the interval $[0, 1]$ while the MDS optimization is running shows transitions between a nonmetric and a metric solution. Moving s temporarily below 1 can help a configuration recover when it gets trapped in a degeneracy (usually clumping of the points in a few locations and near-zero Stress, see Borg and Groenen 1997 sections 13.2-3). Finally, note that the power exponent p cannot be changed in nonmetric mode; it is inherited from the last visit of metric mode.

- A parameter specific to distance scaling, both metric and nonmetric, is the distance power q. We introduced this parameter to include so-called SStress, which is obtained for $p = q = 2$. That is, SStress fits squared distances to squared dissimilarities. SStress is used in the influential MDS software "ALSCAL" by Takane et al. In our limited experience SStress is somewhat less robust than Stress, as the former is even more strongly influenced by large dissimilarities than Stress.

- More esoteric is the choice of metric in configuration space for distance scaling. We allow Minkowski (also called Lebesgue) metrics other than Euclidean by permitting us to manipulate the Minkowski parameter m. The default is Euclidean, $m = 2$; the City block or L_1 metric is the limiting case $m \to 1$.

4.4 Subsetting

The cost functions permit missing values in the dissimilarity matrix: Missing pairs (i, j) are simply dropped from the summations in the cost function and its gradient. Through the deliberate use of missing values users can implement certain extensions of MDS such as multidimensional unfolding (see Borg and Groenen 1997, chapter 14, in particular their figure 14.1).

Missing values can be coded in the dissimilarities file as NAs, or they can be introduced through conditions that are under interactive control. Here is a symbolic description of the summation set of Stress and Strain:

$$I \;=\; \{\; (i,j) \mid i \neq j,\; D_{i,j} \neq NA,\; T_0 \leq D_{i,j} \leq T_1,\;\; \text{Runif}(i,j) < \alpha,\; ... \;\}$$

$0 \leq T_0 \leq T_1$, thresholds, defaults: $T_0 = 0,\; T_1 = \infty$

Runif(i, j) = uniform random numbers $\in [0, 1]$.
α = selection probability, default: $\alpha = 1$.

... = conditions based on color/glyph groups.

Here are details on removal conditions under interactive control:

- **Thresholding:** The lower and upper threshold parameters T_0 and T_1 for the conditions $T_0 \leq D_{i,j} \leq T_1$ are controlled by grips under the histogram in the very bottom of the XGvis control panel (on the left of Figure 3). Thresholding can be used to check the influence of large and small dissimilarities by removing them. We implemented these operations based on the received wisdom that the global shape of MDS configurations is mostly determined by the large dissimilarities. This statement is based on a widely cited study by Graef and Spence (1979) who ran simulations in which they removed, respectively, the largest third and the smallest third of the dissimilarities. They found devastating effects when removing the largest third, but relatively benign effects when removing the smallest third. With interactive thresholding the degree to which this behavior holds can be explored for every dataset individually.

- **Random selection** is implemented by thresholding uniform random numbers $\mathrm{Runif}(i,j)$. The condition $\mathrm{Runif}(i,j) < \alpha$ removes the dissimiliarity $D_{i,j}$ with probability $1-\alpha$. The selection probability α can be controlled interactively (with a slider below the label "Select'n prob", see Figure 3). Because the selection is probabilistic, the number of selected dissimilarities is random (it is shown in a field to the left of the slider). Repeatedly generating new sets of random numbers (by clicking the "new" button) while optimization is continuously running, one gets a sense of how (un)stable the configuration is under random removal of dissimilarities. In our experience classical scaling does not respond well to the removal of even a small fraction of distances. Distance scaling is considerably more robust in this regard.

Another set of removal conditions for dissimilarities is based on color/glyph groups. Such groups can be entered from files or they can be generated interactively with brushing operations in XGobi. We implemented the following ways of using groups in scaling:

- **Subsetting objects:** Remove some objects and scale the remaining objects. Removal is achieved by "hiding" color/glyph groups (click on "Groups" for a menu of the groups and select the "hide" check boxes for the groups that should be removed).
- **Within-groups scaling:** Remove the dissimilarities that cross color/glyph groups. This option can be useful for finding and comparing group-internal structure, which is often obscured in global configurations.

  Within-groups scaling has slightly different behavior in classical and distance scaling: In classical scaling the groups are linked to each other by a common origin, but otherwise they are scaled independently. In distance scaling the groups can be moved independently of each other.

  Note also that nonmetric scaling always introduces a certain dependence between groups because the isotonic transformation is obtained for the pooled within-groups dissimilarities, not for each group separately.
- **Between-groups scaling:** Remove the dissimilarities within the groups. Beetween-groups scaling with two groups is called multidimensional unfolding (Borg and Groenen 1997, chapter 14). Two groups is the case that is most prone to degeneracies because it removes the most dissimilarities. The more groups there are, the more dissimilarities are retained and hence stability is gained.
- **Anchored scaling**: The objects are divided into two subsets, which we call the set of anchors and the set of floaters. We scale the floaters by only using their dissimilarities with regard to the anchors. Floaters are therefore scaled individually, and their positions do not affect each other. The anchors affect the floaters but not vice versa. The configuration of the anchors is dealt with in one of two ways:

– *Fixed anchors:* The anchors have a priori coordinates that determine their configuration. Such coordinates can be entered in an initial position file, or they are obtained from previous configurations by manually moving the anchor points (with mouse dragging).
– *Scaled anchors:* The anchors have dissimilarities also. Configurations for the anchors can therefore be found by subjecting them to regular scaling. Internally scaling the anchors and externally scaling the floaters with regard to the anchors can be done in a single optimization (Section 6.3).

In our practice we usually start with scaled anchors. Subsequently we switch to fixed anchors. Then, while the optimization is running, we drag the anchor points into new locations in order to check the sensitivity and reasonableness of the configuration of the floaters.

The anchor metaphor is ours. Anchored scaling is called "external unfolding" in the literature (Borg and Groenen 1997, Section 15.1).

4.5 Weights

The cost functions are easily adapted to weights. We implemented weights that depend on two parameters, each for a different task:

$$w_{i,j} \;=\; D_{i,j}^r \cdot \begin{cases} w\ , & \text{if color/glyph of } i,j \text{ is the same} \\ (2-w)\ , & \text{if color/glyph of } i,j \text{ is different} \end{cases}$$

$$\begin{array}{ll} -4 \le r \le +4, & r=0: \text{ ignore dissimilarities (default)} \\ & r=-1: \text{ Sammon's mapping} \\ 0 \le w \le 2, & w=1: \text{ ignore groups (default)} \\ & w=2: \text{ within-groups scaling} \\ & w=0: \text{ between-groups scaling} \end{array}$$

The first factor in the weights can depend on a power r of the dissimilarities. If $r > 0$, large dissimilarities are upweighted; if $r < 0$, large dissimilarities are downweighted. This is a more gradual form of moving small and large dissimilarities in and out of the cost function compared to lower and upper thresholding.

For metric distance scaling with $r = -1$, Sammon's mapping (1969) is obtained, an independent rediscovery of a variant of MDS.

The second factor in the weights depends on groups: The parameter w permits continuous up- and downweighting of dissimilarities depending on whether they link objects in the same or different groups. This again is a gradual form of moving between conventional scaling, within-groups scaling, and between-groups scaling. The latter are our ideas, but the weight-based version is due to Priebe and Trosset (2001, forthcoming).

Note that weighting is computationally more costly than subsetting. The latter saves time because some dissimilarities do not need to be looked at, but weights are costly in terms of memory as we store them to save power operations in each iteration.

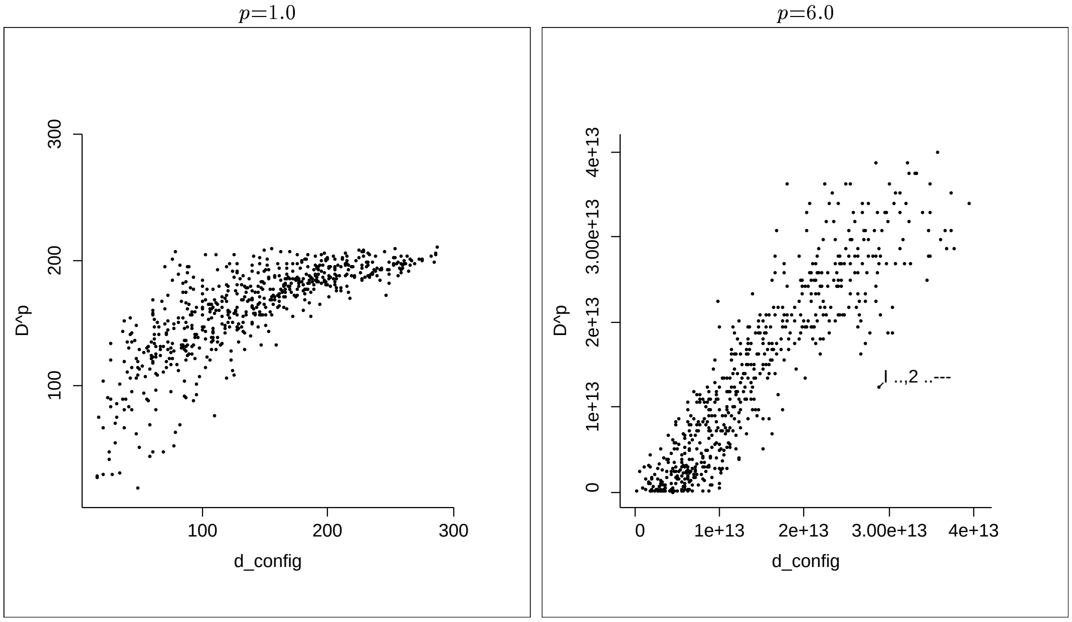


Figure 6: *Two Diagnostic Plots for Configurations from Metric Distance Scaling of the Morse Code Data. Left: raw data, power* $p = 1$*; right: transformed data, power* $p = 6$*. An outlier is marked in the right hand plot: the pair of codes* $(\mathrm{I}, 2) \sim (\cdot\,\cdot\,,\ \cdot\,\cdot - - -)$ *have a fitted distance that is vastly larger than the target dissimilarity.*

5 Diagnostics

A standard diagnostic in MDS is the Shepard plot, which is a scatterplot of the dissimilarities against the fitted distances, usually overlayed with a trace of the isotonic transform. See for example Borg and Groenen (1997, Sections 3.3 and 4.1) or Cox and Cox (1994, Section 3.2.4). The plot provides a qualitative assessment of the goodness of fit, beyond the quantitative assessment given by the Stress value.

The components for a Shepard plot are provided by XGvis on demand (click "Shepard Plot"). The plot is part of a separate XGobi window which goes beyond a mere Shepard plot. The window contains seven variables:

- $d_{i,j}$, the fitted quantities, which are $\|\mathbf{x}_i - \mathbf{x}_j\|$ for distance scaling and $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ for classical scaling (in which case the axis is labeled "b_ij" rather than "d_ij"),
- $f(D_{i,j})$, the transformation of the dissimilarities, which is a power for metric scaling and an isotonic transformation for nonmetric scaling,
- $D_{i,j}$, the dissimilarities,
- $r_{i,j} = f(D_{i,j}) - d_{i,j}$, the residuals,
- $w_{i,j}$, the weights, which may be a power of the dissimilarities,
- i, the row index,
- j, the column index.

Selecting the variables $d_{i,j}$ and $D_{i,j}$ yields the Shepard plot for distance scaling, and an obvious analog for classical scaling. Selecting $D_{i,j}$ and $f(D_{i,j})$ yields a plot of the transformation of the dissimilarities. Selecting $d_{i,j}$ and $f(D_{i,j}$ is possibly even more informative because the strength of the visible linear correlation is a qualitative measure for the quality of the fit. Figure 6 shows an example.

The residuals are provided for users who prefer residual plots over plots of fits. The weights are occasionally useful for marking up- and down-weighted dissimilarities with colors or glyphs. The row and column indices are useful for those XGvis modes in which dissimilarities have been removed from the Stress or Strain, or if some dissimilarities are missing. Plotting i versus j provides a graphical view of the missing and removal pattern of the dissimilarity matrix as it is used in the current cost function.

If the objects are given labels in an input file, the labels of the diagnostics window are pairs of object labels. In Figure 6, for example, an outlying point is labeled "I..,2..---", showing two Morse codes, ".." for "I" and "..---" for "2", whose dissimilarity is not fitted well.

6 Computational and Systems Aspects

6.1 Coordination and Linking of Windows

When starting up XGvis, two windows appear: the XGvis control panel and the XGobi window for the display of the configuration. A diagnostics window for the Shepard plot is created later at the user's discretion.

Both the XGvis panel and its subordinate XGobi display live in the same process. As a result, the XGvis "master" and its XGobi "slave" can trivially share each other's data. For example, the XGvis master is able to feed the XGobi slave the coordinates of the configurations at a fast pace so as to achieve algorithm animation. Similarly, the XGvis master is able to fetch the glyphs and colors from the XGobi slave in order to perform "within-groups MDS," for example.

The window containing the configuration display is a complete XGobi window with all its capabilities for multivariate data visualization, including brushing and labeling (possibly linked to XGobi windows of covariates), data rotations (3-D rotations, grand tours and projection pursuit), parallel coordinate displays, and more (Swayne et al. 1998). We mentioned manually moving points in Section 3: this capability works even when viewing data with 3-D rotations and higher-D grand tours, in which case motion in data space is performed parallel to the current projection plane, leaving the orthogonal backdimensions fixed.

The diagnostics window is a full XGobi window of its own, with data suitable for a Shepard plot among other things. The diagnostics window is *not* linked to the configuration display. In fact, once started up, the window lives in its own independent process, detached from the XGvis process. Users can create diagnostics windows in arbitrary numbers and compare them with each other. The diagnostics window is a frozen snapshot and does not animate itself when XGvis performs optimization. The main reason for detaching the diagnostics window from the rest of XGvis is that the size of its data is in the order N^2 if the size of the configuration is N. A Shepard plot can hence get so large even for moderate N

that the system would slow down to an intolerable degree if animation were attempted. The current number of dissimilarities is shown in the XGvis panel; the display of the number is continuously updated when users perform operations that remove dissimilarities.

6.2 Optimization and Interactivity

For minimization of Stress and Strain we mostly follow Kruskal's (1964b) gradient method. This choice was based on the simplicity with which gradient methods generalize to cost functions with arbitrary weights and non-Euclidean Minkowski metrics. An additional argument could be derived from evidence that the more popular SMACOF algorithm (Borg and Groenen 1997, and references therein) may suffer from premature stopping or stopping at stationary rather than locally optimal solutions (Kearsley, Tapia and Trosset, 1998).

For metric scaling, Kruskal's method is plain gradient descent on the cost function with regard to the configuration. For nonmetric scaling, Kruskal's method consists of alternation between gradient descent on the configuration and estimation of the isotonic transformation. The latter is based on a convex least squares problem that can be solved with Kruskal's pool-adjacent-violators algorithm.

An important part of Kruskal's method is a stepsize strategy. This, however, we do not use because the stepsize in XGvis is under interactive user control. A slider lets users choose the stepsize in relation to the size of the configuration. For example, a stepsize of 0.05 means that the gradient is 5% the size of the configuration, where size of the configuration and the gradient is measured as the sum of the distances from their respective mean vectors.

As we indicated earlier (Section 3), XGvis has no automated stopping criterion either: the system runs a gradient descent loop until the user stops it interactively. An advantage of interactive over automated stopping is that non-trivial local movement in the configuration may still be visually apparent even when the descent in the cost function has become negligible. Another advantage is enabling the "fluid mode of operation" in which the user is free to manipulate configurations and parameters without having to restart optimization after every manipulation.

6.3 Gradient Descent on the Configuration

We now focus on details of the gradient descent step with a fixed transformation. The point of doing so is to bring conventional MDS and anchored MDS into a single computational framework.

We first note that minimizing Stress and Strain is equivalent to maximizing the respective cosine expressions of sections 4.1 and 4.2. Omitting terms that do not depend on the configuration, we are to maximize a ratio of the following form:

$$\frac{Num}{Denom} = \frac{\sum_{(i,j)\in I} \; w_{i,j} \cdot g(D_{i,j}) \cdot s(\mathbf{x}_i, \mathbf{x}_j)}{\left(\sum_{(i,j)\in I} \; w_{i,j} \cdot s(\mathbf{x}_i, \mathbf{x}_j)^2\right)^{1/2}} \, ,$$

where $s(\mathbf{x}_i, \mathbf{x}_j)$ is a Minkowski distance (to the q'th power) for distance scaling, and the inner product for classical scaling. Both, distances and inner products, are symmetric in their arguments, hence $s(\mathbf{x}_i, \mathbf{x}_j)$ is symmetric in i and j. As $D_{i,j}$ is assumed symmetric in the subscripts, so is $w_{i,j}$.

The gradient of the ratio with regard to the configuration $\mathbf{X} = (\mathbf{x}_i)_{i=1..N}$ is the collection of partial gradients with regard to configuration points $\mathbf{x}_i$:

$$\frac{\partial}{\partial \mathbf{X}} \frac{Num}{Denom} = \left(\frac{\partial}{\partial \mathbf{x}_i} \frac{Num}{Denom} \right)_{i=1..N} .$$

Because the gradient is sized in relation to the configuration, we only need the partial gradients up to factors that do not depend on i:

$$\frac{\partial}{\partial \mathbf{x}_i} \frac{Num}{Denom} \propto \sum_{j:(i,j)\in I} w_{i,j} \left(g(D_{i,j}) - \frac{Num}{Denom^2}\, s(\mathbf{x}_i, \mathbf{x}_j) \right) \frac{\partial}{\partial \mathbf{x}} \bigg|_{\mathbf{x}=\mathbf{x}_i} s(\mathbf{x}, \mathbf{x}_j) ,$$

In the derivation we used symmetry of $D_{i,j}$, $w_{i,j}$ and $s(\mathbf{x}_i, \mathbf{x}_j)$ in i and j. The summation should really extend over the set

$$\{j \,|\, (i,j) \in I \text{ or } (j,i) \in I \} ,$$

but if I is also symmetric, that is, $(i,j) \in I \Rightarrow (j,i) \in I$, then it is sufficient to sum over the reduced set

$$\{j \,|\, (i,j) \in I\}$$

This is the gradient computation implemented in XGvis.

Partial gradients have an intuitive interpretation: The summand indexed by j is the contribution of $\mathbf{x}_j$ to the partial gradient of the point $\mathbf{x}_i$. As such, it can be interpreted as the "force" exerted by $\mathbf{x}_j$ on $\mathbf{x}_i$. With this metaphor the reduced summation can be cast as follows: Both $D_{i,j}$ and $D_{j,i}$ exert force on $\mathbf{x}_i$, but the symmetry assumptions allow us to use $D_{i,j}$ only. The gradient computed with reduced summation will be called "reduced gradient" for short.

Iterations based on the reduced gradient solve really a problem that is conceptually different from Stress/Strain minimization. To see this, we introduce "bimodal" Stress/Strain, or a "bimodal" $Num/Denom$ ratio, as follows:

$$\frac{Num}{Denom}(\xi_1, ..., \xi_N | \mathbf{x}_1, ..., \mathbf{x}_N) = \frac{\sum_{(i,j)\in I} \; w_{i,j} \cdot g(D_{i,j}) \cdot s(\xi_i, \mathbf{x}_j)}{\left(\sum_{(i,j)\in I} \; w_{i,j} \cdot s(\xi_i, \mathbf{x}_j)^2 \right)^{1/2}}$$

Maximizing this with regard to the ξ_i's given fixed $\mathbf{x}_j$'s amounts to anchored scaling with fixed anchors $\{\mathbf{x}_1, ..., \mathbf{x}_N\}$ and floaters $\{\xi_1, ..., \xi_N\}$. Our reduced gradient is exactly the plain gradient of $Num/Denom(\xi_1, ..., \xi_N | \mathbf{x}_1, ..., \mathbf{x}_N)$ with regard to the ξ_i's at $\xi_i = \mathbf{x}_i \; \forall i$. Reduced gradient steps therefore try to find fixed configurations under the mapping

$$(\mathbf{x}_1, ..., \mathbf{x}_N) \mapsto (\xi_1, ..., \xi_N) = \text{argmax}_{(\xi_1, ..., \xi_N)} \frac{Num}{Denom}(\xi_1, ..., \xi_N | \mathbf{x}_1, ..., \mathbf{x}_N) .$$

We know from the earlier argument that reduced gradient steps do find local maxima of the "unimodal" ratio $Num/Denom\,(\mathbf{x}_1, ..., \mathbf{x}_N)$ when the summation set I is symmetric. Now, the new "bimodal" interpretation allows us to incorporate anchored scaling in the same framework with summation sets I that are *not* symmetric.

We illustrate this point with an example. Consider a dissimilarity matrix whose first column is missing: $D_{i,1} = NA \ \forall i$, or equivalently, $I = \{(i,j) \,|\, 1 \le i \le N,\ 2 \le j \le N\}$. We note that $\mathbf{x}_1$ does not contribute to the reduced partial gradient of any point whatsoever, but its reduced partial gradient has contributions from those points for which $D_{1,j}$ is not missing. Intuitively, $\mathbf{x}_1$ does not exert force but it "feels" force from other points; it does not "push", but it is being "pushed". In other words, $\mathbf{x}_1$ is a floater, and its anchors are $\{\mathbf{x}_j \,|\, D_{1,j} \ne NA\}$. This example shows that a suitable NA pattern in the dissimilarities permits us to implement anchored scaling in addition to conventional scaling.

We discuss briefly the NA patterns of the group-based scaling methods that are provided as interactive selections in XGvis (Section 4.4):

- Within-groups scaling, illustrated with two groups:

$$D \;=\; \begin{array}{cc} \begin{array}{cc} \text{group1} & \text{group2} \end{array} & \\ \begin{pmatrix} D_{grp1,grp1} & NA \\ NA & D_{grp2,grp2} \end{pmatrix} & \begin{array}{l} \text{group1} \\ \text{group2} \end{array} \end{array}$$

  Group 1 gets scaled internally, and so does group 2. The forces are confined within the groups. The summation set I is symmetric.

- Between-groups scaling, again illustrated with two groups (multidimensional unfolding):

$$D \;=\; \begin{array}{cc} \begin{array}{cc} \text{group1} & \text{group2} \end{array} & \\ \begin{pmatrix} NA & D_{grp1,grp2} \\ D_{grp2,grp1} & NA \end{pmatrix} & \begin{array}{l} \text{group1} \\ \text{group2} \end{array} \end{array}$$

  The two groups exert force on each other, but there is no force within the groups. The summation set I is again symmetric.

- Anchored scaling with scaled anchor (a form of external unfolding):

$$D \;=\; \begin{array}{cc} \begin{array}{cc} \text{anchors} & \text{floaters} \end{array} & \\ \begin{pmatrix} D_{ancr,ancr} & NA \\ D_{fltr,ancr} & NA \end{pmatrix} & \begin{array}{l} \text{anchors} \\ \text{floaters} \end{array} \end{array}$$

  Here the summation set I is no longer symmetric. The top left submatrix causes conventional scaling of the anchors. The bottom left submatrix exerts the push of the anchors on the floaters. The two blocks of NA's on the right imply that the columns for the floaters are absent, that is, the floaters do not push any points.

- Anchored scaling with fixed anchor (another form of external unfolding):

$$D \;=\; \begin{array}{cc} \begin{array}{cc} \text{anchors} & \text{floaters} \end{array} & \\ \begin{pmatrix} NA & NA \\ D_{fltr,ancr} & NA \end{pmatrix} & \begin{array}{l} \text{anchors} \\ \text{floaters} \end{array} \end{array}$$

Again the summation set I is not symmetric. The two top blocks of NA's imply that the anchors are fixed: they are not being pushed by anyone. The only push is exerted by the anchors on the floaters through the matrix on the bottom left.

It becomes clear that NA patterns form a language for expressing arbitrarily complex constellations of forces. This idea can be formalized in terms of what we may call a "force graph", defined as the directed graph with nodes $\{1, ..., N\}$ and edges in the summation set

$$I = \{(i,j) \mid D_{i,j} \neq NA\} .$$

An edge (i,j) stands for "j pushes i". Conventional MDS is represented by a complete graph, where every point pushes every other point. For within-groups scaling the force graph decomposes into disconnected complete subgraphs (cliques). Between-groups scaling has a complete bi-directional bipartite graph, that is, the node set is decomposed into two disjoint partitions, and the edge set is the set of edges from either partition to the other. Anchored MDS with fixed anchors has a uni-directional complete bipartite graph, that is, the two partitions have asymmetric roles in that the edges go only from one of the partitions to the other. In anchored MDS with scaled anchors the latter form in addition a clique. One can obviously conceive of more complex force graphs, such as multi-partite graphs with layered anchoring, or graphs with selected force cycles, but this is not the place to pursue the possibilities in detail.

6.4 MDS on Large Numbers of Objects

MDS is based on N^2 algorithms, a fact that limits its reach for large N. On the hardware at our disposal, interactive use of XGvis is possible up to about $N = 1000$. Larger N can be processed also, but the user will leave the optimization to itself for a while. The largest N we have scaled with XGvis was $N = 3648$, but still larger N are feasible with more patience.

Among the four types of MDS, the nonmetric varieties are never recommended for N greater than a few hundred because setting up the isotonic regression adds initially a considerable computational burden. Among the two metric varieties, classical scaling is faster than distance scaling. It is therefore a common strategy to use classical scaling solutions as initial configurations for distance scaling. For what it's worth, on a laptop computer with model year 2000 the data example of Section 7.2 with $N = 1926$ required about 5 seconds per gradient step for metric distance scaling and under 4 seconds per step for metric classical scaling.

We mentioned in Section 4.5 that weighted MDS is costly. When N is large, one should abstain from the use of weights for space reasons. XGvis is programmed not to allocate a weight array if the weights are identical to 1.

The reach of MDS extends when a substantial number of terms is trimmed from the Stress function. Such trimming is most promising in anchored MDS (Section 4.4), which can be applied if an informative set of anchors can be found. The choice of anchors can be crucial; in particular, a random choice of anchors often does not work. But we have had success with the example of size $N = 3648$ mentioned earlier: satisfying configurations were found with an anchor set of size 100, which reduced the time needed for a single gradient step from 100 seconds to 6 seconds.

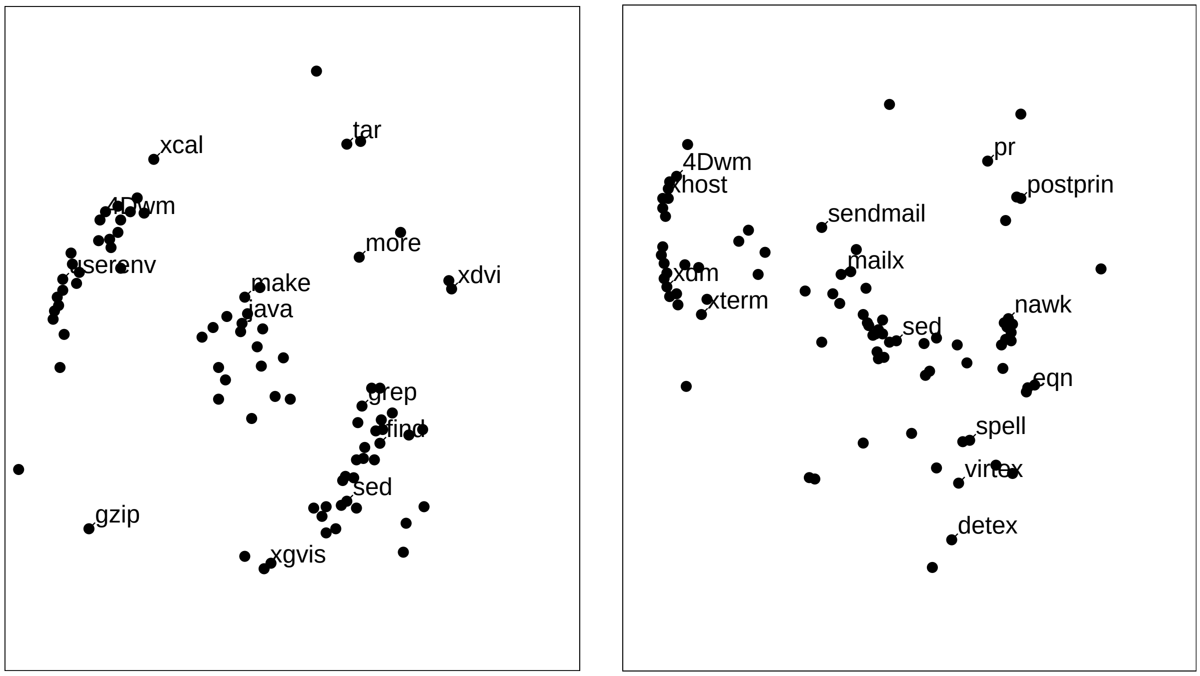


Figure 7: *Maps of Computer Commands for Two Individuals.*
Left: a statistician who does programming and data manipulation (Stress=0.29).
Right: an administrative assistant who does e-mail and word processing (Stress=0.34).

7 A Tour of MDS Applications

We give a tour of data examples that demonstrate the wide applicability of MDS and the usefulness of XGvis.

7.1 Distance Data for Computer Usage

A type of data that poses difficulty for visualization (or any statistical analysis for that matter) are non-rectangular data such as transactional data and event data, where objects of interest are characterized by a variable number of transactions or events. Examples are phone call data that characterize phone company customers, or earthquake data that characterize geographic areas. The typical approach to such data is to extract a fixed number of features for each object, thus transforming non-rectangular to rectangular multivariate data. Another, lesser known, approach is extraction of distance data: Often it is not too difficult to directly define sensible distance measures between series of transactions and events. Given a distance matrix for all pairs of objects, MDS becomes the tool of choice for visualization.

We illustrate the idea with computer usage data collected at AT&T Labs as part of a project on intrusion detection for computer networks (Theus and Schonlau 1998, Dumouchel and Schonlau 1998). The purpose of this particular data collection was to examine behavioral indicators that would discriminate between known, legitimate users. The data shown here are courtesy of Schonlau and Theus.

The data are essentially a record of all operating system commands executed by a given user over the course of several days. Theus and Schonlau (1998) explored individual usage patterns by calculating for a given individual a set of distance measures for commands: two commands are close for this individual if he/she often uses the commands close to each other in time. The similarity measure they used was roughly speaking the number of times two commands appeared adjacent in the log file. The resulting similarities were converted to distances, which in turn were used to create a map of the commands with MDS. The hope was that users would exhibit strong individual signatures in these maps, which they did indeed.

Figure 7 shows maps for two individuals, one a statistician, the other an administrative assistant. Both maps show in the upper left a chain of commands that originates from the startup of a session, when among other things the window manager ("4Dwm") is being launched. Otherwise the two maps are very different: The statistician on the left shows a cluster in the center arising from programming activity, and a cluster in the lower right arising from data manipulation and use of XGvis. The administrative assistant shows some e-mail activity near the startup sequence and word processing work in the right half of the map. These two individuals are somewhat extreme in their differences, but maps of more homogeneous groups of users carry strong individual signatures also.

7.2 Dimension Reduction

In order to use MDS for dimension reduction, one computes a distance matrix from a high-dimensional multivariate dataset and subjects the matrix to MDS in lower dimensions, such as two or three. The potential advantage of MDS over other methods such as principal components analysis (PCA) is its nonlinearity and consequent greater flexibility. A disadvantage of MDS sometimes mentioned is the lack of a mapping between data space and configuration space: MDS maps only the observations, not the embedding space, whereas PCA produces projections that are defined on all of data space. This argument against MDS can be countered to some extent: an embedding of new test data is easily achieved with MDS anchored on the training data. The remaining disadvantage of MDS is the potential non-uniqueness of mapping with anchored MDS, which is the price of nonlinearity.

Our data example is from a market segmentation study at AT&T. Analysts constructed four market segments by applying k-means clustering to roughly 2,000 households characterized by eight variables. The data do not fall into natural clusters, and the k-means algorithm simply slices the data like a pie. The four k-means centers lie very near the plane spanned by the first two principal components, resulting in the coherent Voronoi regions visible in the PCA projection in the right hand plot of Figure 8.

The result of metric MDS is shown in the left hand plot in Figure 8: The point scatter is very similar to the PCA projection, although more rounded on the periphery and less crowded in the center. The Stress value was 0.25, indicating substantial residual structure. By comparison, the first two principal components explained 55% of the variance, which translates to a Stress of 0.45. Thus, MDS squished 20% more of the variance into two dimensions compared to PCA. This "success" comes at a cost: Part of the additional variance is probably noise, and noise causes artifacts in MDS, namely, a rounded periphery and

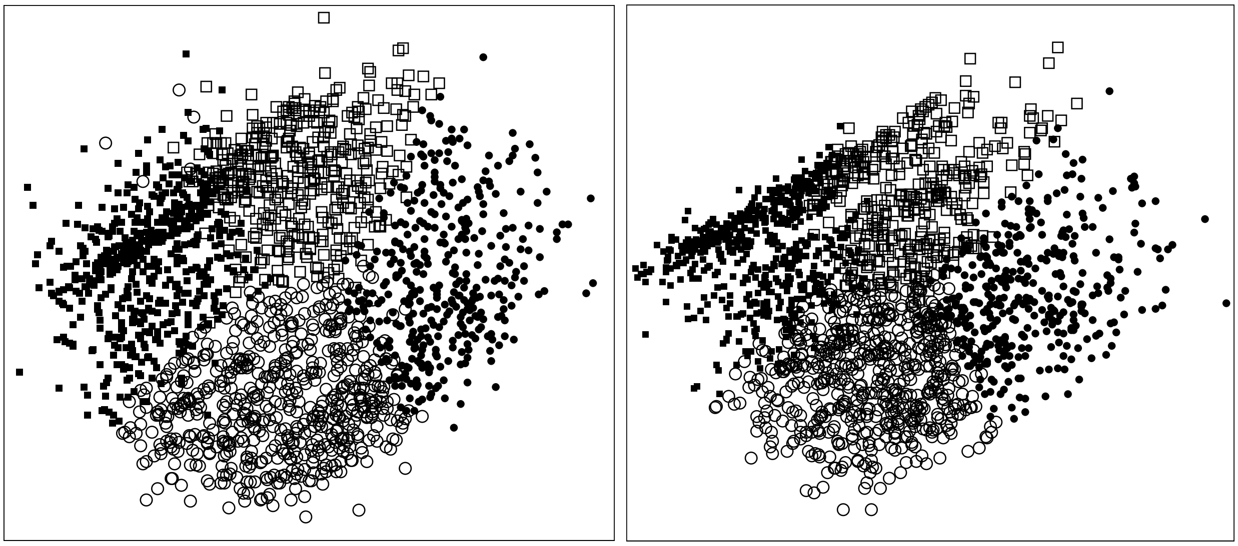


Figure 8: *Marketing Segmentation Data. Left: MDS reduction to 2-D; right: largest two principal components. The glyphs represent four market segments constructed with k-means clustering using four means.*

decreased mass in the center (Buja et al. 1994). Both effects are present in the MDS configuration of Figure 8. Thus the tradeoff is higher Stress values in PCA versus mild artifacts in MDS.

Note that the PCA projection can in principle be obtained with metric classical scaling in XGvis, but working with a 1926×1926 distance matrix is inefficient when an eigendecomposition of an 8×8 matrix achieves the same result.

7.3 Graph Layout: Telephone Call Graphs

The development of XGvis was originally motivated by the problem of laying out graphs in more than two dimensions and using XGobi as a display device (Littman et al. 1992). We first used a "spring model" (Di Battista et al. 1994) for graph layout but soon found that MDS is superior in that it gets trapped less often in messy local minima and more often finds globally meaningful topologies. There might, however, exist a role for spring models as post-processors of MDS layouts (L. Wilkinson, personal communication).

The use of more than two dimensions for graph layout is an obvious way of avoiding edge crossings, the bane of graph layout in two dimensions. Strictly speaking, edge crossings are not avoided by laying out in 3-D: the drawing plane is still 2-D, but the 2-D display of a moving layout in 3-D creates the illusion of avoiding edge crossings. Another way of looking at 3-D layout is the following: Rotating a 3-D layout and projecting it to 2-D amounts to moving among a 2-parameter family of 2-D layouts. Rotation then allows users to search the 2-parameter family and find the layout that best reveals local detail of interest.

In order to lay out a graph with MDS, one computes from the graph a distance matrix that can be subjected to MDS. The simplest notion of distance in graphs is the so-called "shortest-path metric": for each pair of nodes, define their distance to be the length of the

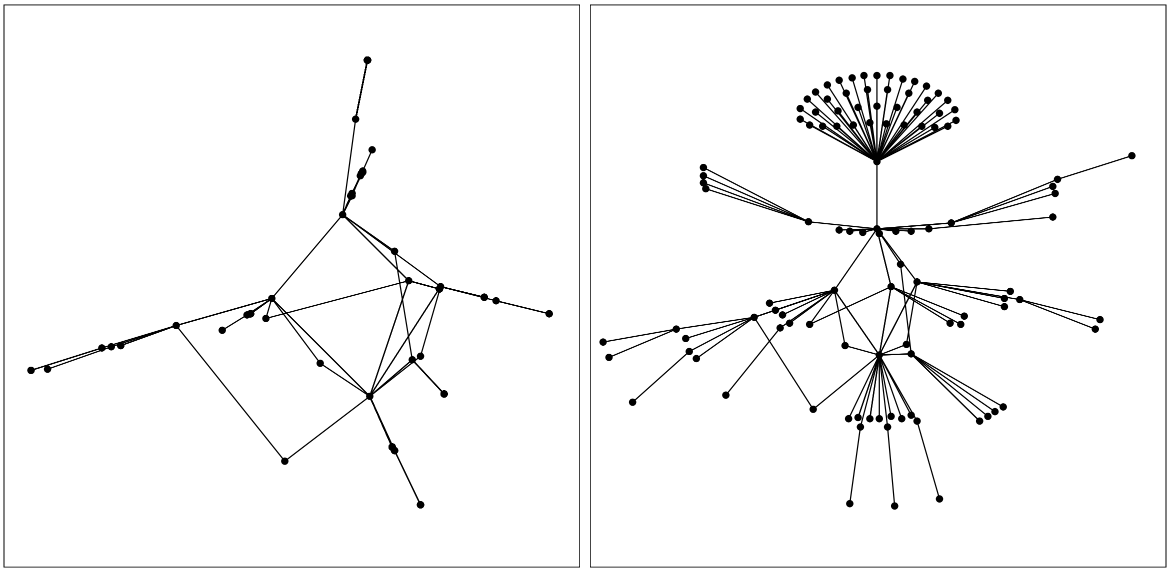


Figure 9: *A Telephone Call Graph, Layed Out in 2-D. Left: classical scaling (Stress=0.34); right: distance scaling (Stress=0.23). The nodes represent telephone numbers, the edges represent the existence of a call between two telephone numbers in a given time period.*

shortest path between the two nodes. XGvis accepts graphs as input data, represented as a list of pairs of object numbers. XGvis computes the shortest-path metric at start-up. It also represents the graph visually with lines connecting configuration points.

XGvis does not address very large graphs. The largest problem we have successfully scaled had 3648 objects. Full MDS took hours, but anchored MDS made the optimization quite doable. For a system that can lay out very large graphs in 2-D by different means, see Wills' "NicheWorks" (1999).

Our data example is a particular type of social network: a telephone call graph in which telephone numbers are nodes and the existence of calls made between two telephone numbers constitutes an edge. In what follows we treat the graph as undirected, ignoring the directionality of caller-callee. Call graphs as seen by telephone companies tend to be extremely large; the techniques described in this paper are applicable only to relatively small subgraphs. For the following example, we started with a call graph consisting of tens of millions of nodes. This large graph was broken up into strongly connected components (Abney et al. 1998). Because we had no good prior intuitions about the structure of call graphs, we examined a few of these components visually in XGvis. Figure 9 shows one example, a connected component with 110 nodes.

The comparison of classical and distance scaling in Figure 9 shows that the former can be problematic for graphs: it tends to collapse dangling subtrees. Distance scaling, however, is able to spread these subtrees out to generate a "dandelion" effect. (Compare de Leeuw's (1984) result that locally optimal solutions of distance scaling generally avoid ties in the configuration.) The advantage of classical scaling is that it shows the more heavily connected parts of the graph more prominently.

A useful strategy for laying out larger graphs is to first apply classical scaling for speed,

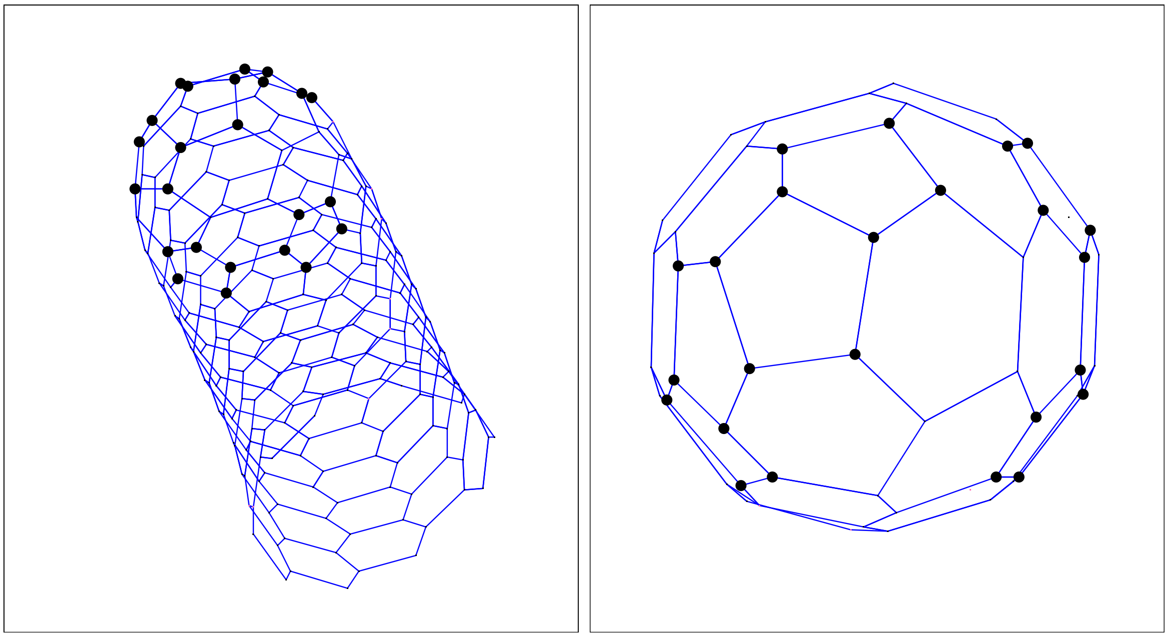


Figure 10: *Nanotube Embedding. One of Asimov's graphs for a nanotube is rendered with MDS in 3-D (Stress=0.06). The nodes represent carbon atoms, the lines represent chemical bonds. The right hand frame shows the cap of the tube only. Some of the pentagons are clearly visible.*

followed by a few gradient steps of distance scaling. To untangle the dangling subtrees with distance scaling, one may have to first perturb the configuration with a small amount of noise.

As for insights that can be gained from the layouts: it turns out that the dandelion in Figure 9 arises from a hotel. This was a valuable negative finding which led us to refine the definition of the call graph by eliminating uninteresting nodes.

7.4 Molecular Conformation: Embedding of Nanotubes in 3-D

The following is an application of MDS to the geometric reconstruction of so-called nanotubes. Nanotubes are large carbon molecules in which the carbon atoms are arranged so as to form tubes that may have open ends or capped ends. The electrical and physical properties of these tubes may spur promising new technologies. [An account for the lay reader appeared in the New York Times, February 17, 1998, "Science Times" p. C1,5, complete with a photo of a wire model.] The geometry of capped nanotubes was described by D. Asimov (1998), who developed a procedure for constructing all possible cappings of all possible tubes. The results of his constructions are graphs in which the nodes correspond to carbon atoms and the edges correspond to chemical bonds. Given a graph representing the intrinsic structure of a nanotube, one is left with the task of embedding the graph in 3-space in order to obtain a full geometric representation of a nanotube. Such an embedding can be achieved with MDS.

We generated according to Asimov's theory a graph with 222 nodes, 549 edges, representing a half-tube with a cap on one end and an opening on the other end. Subjecting the graph to metric distance scaling in XGvis produced the 3-D embedding shown in Figure 10. Clearly visible is the hexagonal tiling structure of the graph. The notable exception to the hexagonal structure is in the cap where exactly six pentagons can be found; they permit the necessary bending of the cap. These pentagons are made visible with highlighted points.

We were surprised by the good quality of the nanotube embedding in view of the fact that the shortest-path metric derived from a graph is not Euclidean. Surprisingly, the deviation from the Euclidean ideal turned out to be minor: The Stress of the final embedding was a low 0.06. An obvious minor imperfection is the slight widening of the uncapped end of the half-tube. In chemical applications one will use the MDS configuration as input to a chemical bond optimizer which can achieve much greater precision.

8 Conclusions

This article has several purposes:

- It gives an introduction to multidimensional scaling from the point of view of interactive data visualization.
- It describes a system, called XGvis, that acts as a test bed for exploring the reach of human control in MDS computations.
- It outlines an extensive set of tools for creating, visualizing, manipulating and diagnosing MDS configurations.
- It gives a selected tour of modern MDS applications.

We hope to have shown that MDS as a data visualization technique is tailor-made for contemporary interactive computing environments and of great use in some important and attractive data problems. The XGvis/XGobi software is freely available from the following web site:

http://www.research.att.com/areas/stat/xgobi/

The software runs under the X Window System™. It has been used on several types of UNIX®and Linux®systems. Brian Ripley has also ported the software to Microsoft Windows™(an X emulator is required):

http://www.stats.ox.ac.uk/pub/SWin/xgobi.zip

Acknowledgments

We wish to thank Jon Kettenring and Daryl Pregibon, the two managers who made this work possible. We are indebted to Brian Ripley for his port of the software to Windows.

X Window System is a trademark of MIT.
UNIX is a registered trademark of The Open Group.
Linux is a registered trademark of Linus Torvalds.
Microsoft Windows is a trademark of Microsoft, Inc.

Selected Literature

Multidimensional scaling is the subject of several current books: Borg and Groenen (1997), Cox and Cox (1994), Young and Hamer (1994), Davison (1992), as well as some older ones: Borg and Lingoes (1987), Schiffman (1981), Kruskal and Wish (1978). In this article we made heavy use of Borg and Groenen's book. At the time of writing it is the most recent as well as the broadest in its coverage not only of algorithms but also of general methodology. By comparison the book by Cox and Cox recommends itself for a compact survey of a large literature.

The collection edited by Davies and Coxon (1982) contains some of the seminal articles in the field, including Kruskal's (1964a,b). An older book article we still recommend is Greenacre and Underhill (1982). Many books on multivariate analysis include chapters on multidimensional scaling, such as Gnanadesikan (1997) and Seber (1984).

The area of MDS is currently dominated by two groups of authors: the "Dutch school" of Heiser, Meulman, Groenen (see his 1993 thesis), de Leeuw and their co-workers; and Arabie, Carroll, Hubert and co-workers. Any study of recent developments in the field should start with their work.

References

[1] Abney, S., Volinsky C., and Buja A. (1998), "Mining Large Call Graphs," AT&T Labs Technical Memorandum.

[2] Asimov, D. (1998), "Geometry of Capped Nanocylinders," AT&T Labs Technical Memorandum, http://www.research.att.com/areas/stat/nano

[3] Carroll, J. D., and Arabie, P. (1998), "Multidimensional scaling," in M. H. Birnbaum (Ed.), *Handbook of Perception and Cognition, Volume 3: Measurement, Judgment, and Decision Making*, San Diego, CA: Academic Press, pp. 179-250.

[4] Borg, I., and Groenen, P. (1997), *Modern Multidimensional Scaling: Theory and Applications*, Springer Series in Statistics, New York: Springer-Verlag.

[5] Borg, I., and Lingoes, J. (1987), *Multidimensional Similarity Structure Analysis*, New York: Springer-Verlag.

[6] Buja, A., Cook, D., and Swayne, D. F. (1996), "Interactive high-dimensional data visualization," *Journal of Computational and Graphical Statistics*, **5**, pp. 78–99. A companion video tape can be borrowed from the ASA Statistical Graphics Section lending library.

[7] Buja, A., Logan, B. F., Reeds, J. R., Shepp, L. A. (1994), "Inequalities and positive-definite functions arising from a problem in multidimensional scaling," *The Annals of Statistics*, **22**, pp. 406-438.

[8] Cox, R. F., and Cox, M. A. A. (1994), *Multidimensional Scaling*, Monographs on Statistics and Applied Probability, London: Chapman & Hall.

[9] Crippen, G. M., and Havel, T. F. (1978), "Stable calculation of coordinates from distance information," Acta crystallographica, **A34**, pp. 282-284.

[10] Davison, M. L. (1992), *Multidimensional Scaling*, Melbourne, Florida: Krieger Publishing Company.

[11] Davies, P. M., and Coxon, A. P. M. (eds.) (1982), *Key Texts in Multidimensional Scaling*, Exeter, New Hampshire: Heinemann Educational Books Inc.

[12] de Leeuw, J. (1984), "Differentiability of Kruskal's Stress at a Local Minimum," *Psychometrica*, **49**, pp. 111-113. Fairfax Station, VA: Interface Foundation of North America, Inc., pp. 208–217.

[13] Di Battista, G., Eades, P., Tamassia, R. and Tollis, I. (1994), "Algorithms For Drawing Graphs: An Annotated Bibliography," *Computational Geometry*, **4**, pp. 235-282.

[14] DuMouchel, W. and Schonlau, M. (1998), "A Comparison of Test Statistics for Computer Intrusion Detection Based on Principal Components Regression of Transition Probabilities," *Proceedings of the 30th Symposium on the Interface: Computing Science an d Statistics.* Fairfax Station, VA: Interface Foundation of North America, Inc.

[15] Glunt, W., Hayden, T. L., and Raydan, M. (1993), "Molecular conformation from distance matrices," Journal of Computational Chemistry **14** 1, pp. 114-120.

[16] Gnanadesikan, R. (1997), *Methods for Statistical Data Analysis of Multivariate Observations*, Wiley Series in Probability and Statistics, New York: John Wiley & Sons.

[17] Graef J., and Spence, I. (1979), "Using Distance Information in the Design of Large Multidimensional Scaling Experiments," Psychological Bulletin **86**, pp 60-66.

[18] Greenacre, M. J., and Underhill, L. G. (1982), "Scaling a Data Matrix in a Low-Dimensional Euclidean Space," in *Topics in Applied Multivariate Analysis*, Hawkins, D. M., ed., Cambridge UK: Cambridge University Press, pp. 183-268.

[19] Groenen, P. J. F. (1993), "The Majorization Approach to Multidimensional Scaling: Some Problems and Extensions ," Leiden: DSWO Press.

[20] Havel, T. F. (1991), "An evaluation of computational strategies for use in the determination of protein structure from distance constraints obtained by nuclear magnetic resonance," Progress in Biophysics and Molecular Biology **56**, pp. 43-78.

[21] Kearsley, A. J., Tapia, R. A., and Trosset, M. W. (1998), "The solution of the metric STRESS and SSTRESS in multidimensional scaling by Newton's method," *Computational Statistics*, **13**, pp. 369–396.

[22] Kruskal, J. B., and Wish, M. (1978), *Multidimensional Scaling*, Sage University Paper series on Quantitative Application in the Social Sciences, 07-011. Beverly Hills and London: Sage Publications.

[23] Kruskal, J. B. (1964a), "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," Psychometrika **29**, pp 1-27.

[24] Kruskal, J. B. (1964b), "Nonmetric multidimensional scaling: a numerical method," Psychometrika **29**, pp. 115-129.

[25] Kruskal, J. B., and Seery, J. B. (1980), "Designing Network Diagrams," *Proceedings of the First General Conference on Social Graphics, July 1980,* US Dept of the Census, Washington, DC, pp. 22-50.

[26] Littman, M., Swayne, D. F., Dean, N., and Buja, A. (1992), "Visualizing the embedding of objects in Euclidean space," *Computing Science and Statistics: Proc. of the 24th Symp. on the Interface*, Fairfax Station, VA: Interface Foundation of North America, Inc., pp. 208–217.

[27] McFarlane, M., and Young, F. W. (1994), "Graphical Sensitivity Analysis for Multidimensional Scaling," J. of Computational and Graphical Statistics **3**, pp. 23-33.

[28] Meulman, J.J. (1992), "The integration of multidimensional scaling and multivariate analysis with optimal scaling," Psychometrika **57**, pp. 539–565.

[29] Rothkopf, E. Z. (1957), "A measure of stimulus similarity and erroros in some paired-associate learning tasks," Journal of Experimental Psychology **53**, pp. 94-101.

[30] Sammon, J. W., (1969), "A Non-Linear Mapping for Data Structure Analysis," IEEE Trans. on Computers, C-18(5).

[31] Schiffman, S. (1981), *Introduction to Multidimensional Scaling : Theory, Methods, and Applications*, San Diego and London: Academic Press.

[32] Seber, G. A. F. (1984), *Multivariate Observations*, New York: John Wiley & Sons.

[33] Shepard, R. N. (1962), "The analysis of proximities: multidimensional scaling with an unknown distance function," I and II, *Psychometrika*, **27**, pp. 125-139 and pp. 219-246.

[34] Swayne, D. F., Cook, D., and Buja, A. (1998), "XGobi: Interactive Data Visualization in the X Window System," *Journal of Computational and Graphical Statistics*, **7**, pp. 113-130.

[35] Theus, M. and Schonlau, M., (1998), "Intrusion Detection Based on Structural Zeroes," *Statistical Computing & Graphics Newsletter*, Vol. 9, No 1, 12-17, Alexandria, VA: American Statistical Association.

[36] Takane, Y., Young, F. W. and De Leeuw, J. (1977), "Nonmetric individual differences multidimensional scaling: An alternating least-squares method with optimal scaling features," *Psychometrika*, **42**, pp 7-67.

[37] Torgerson, W. S. (1952), *Psychometrika*, 17, pp. 401-419.

[38] Trosset, M. W. (1998a), "Applications of Multidimensional Scaling to Molecular Conformation," Computing Science and Statistics, 29, pp. 148-152.

[39] Trosset, M. W. (1998b), "A New Formulation of the Nonmetric Strain Problem in Multidimensional Scaling," *Journal of Classification*, **15**, pp. 15-35.

[40] Wills G. J. (1999), "NicheWorks - Interactive Visualization of Very Large Graphs," *Journal of Computational and Graphical Statistics* **8**, pp 190-212.

[41] Young, F. W., Hamer, R. M. (1994), *Theory and Applications of Multidimensional Scaling*, Hillsdale, NJ: Eribaum Associates.