Data Visualization with Multidimensional Scaling

Andreas BUJA[1], Deborah F. SWAYNE[2],
Michael L. LITTMAN[3], Nathaniel DEAN[4],
Heike HOFMANN[5], Lisha CHEN[6].

September 18, 2007

We discuss methodology for multidimensional scaling (MDS) and its implementation in two software systems ("GGvis" and "XGvis"). MDS is a visualization technique for proximity data, that is, data in the form of $N \times N$ dissimilarity matrices. MDS constructs maps ("configurations", "embeddings") in $\mathbb{R}^k$ by interpreting the dissimilarities as distances. Two frequent sources of dissimilarities are high-dimensional data and graphs. When the dissimilarities are distances between high-dimensional objects, MDS acts as a (often nonlinear) dimension reduction technique. When the dissimilarities are shortest-path distances in a graph, MDS acts as a graph layout technique. MDS has found recent attention in machine learning motivated by image databases ("Isomap," Tenenbaum et al. 2004). MDS is also of interest in view of the popularity of "kernelizing" approaches inspired by SVMs ("kernel PCA," Schölkopf et al. 1998).

This article discusses the following general topics: (1) the stability and multiplicity of MDS solutions; (2) the analysis of structure within and between subsets of objects with missing value schemes in dissimilarity matrices; (3) gradient descent for optimizing general MDS loss functions ("Strain" and "Stress"); (4) a unification of classical (Strain-based) and distance (Stress-based) MDS.

Particular topics include the following: (1) blending of automatic optimization with interactive displacement of configuration points to assist in the search for global optima,

[1]Andreas Buja is the Liem Sioe Liong / First Pacific Company Professor, The Wharton School, University of Pennsylvania, Philadelphia, PA 19104-6340. (http://www-stat.wharton.upenn.edu/~buja)

[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 Associate Professor, Rutgers, The State University of New Jersey, Department of Computer Science, Hill Center Room 409, Piscataway, NJ 08854-8019
(mlittman@cs.rutgers.edu, http://www.cs.rutgers.edu/~mlittman/)

[4]Nathaniel Dean is Professor, Department of Mathematics, Texas State University - San Marcos, 601 University Drive, San Marcos, TX 78666
(nd17@txstate.edu, http://www.txstate.edu/math/people/faculty/dean.html)

[5]Heike Hofmann is Assistant Professor, Dept of Statistics, Iowa State University, Ames, IA 50011.
(hofmann@iastate.edu, http://www.public.iastate.edu/ hofmann/)

[6]Lisha Chen is Assistant Professor, Department of Statistics, Yale University, 24 Hillhouse Ave, New Haven, CT 06511 (lisha.chen@yale.edu, http://www.stat.yale.edu/~lc436/)

(2) forming groups of objects with interactive brushing to create patterned missing values in MDS loss functions, (3) optimizing MDS loss functions for large numbers of objects relative to a small set of anchor points ("external unfolding"), (4) a nonmetric version of classical MDS.

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.

Key Words: Proximity Data, Dissimilarity Data, Multivariate Analysis, Dimension Reduction, Multidimensional Unfolding, External Unfolding, Graph Layout, Social Networks, Molecular Conformation, gradient descent.

1 Introduction: Basics of Multidimensional Scaling

In this section we give a short introduction to those types of MDS that are relevant for this article. Section 2 gives an overview of interactive MDS operations, as they can be realized in XGvis or GGvis. Section 3 approaches the stability and multiplicity problem of MDS configurations with algorithm animation, direct manipulation and perturbation of the configuration. Section 4 gives details about the loss functions and their parameters for controlling transformations, subsetting and weighting of dissimilarities. Section 5 describes diagnostics for MDS. Section 6 is concerned with 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 higher dimensions. An earlier article by Buja and Swayne (2002) gives further illustrations of the methodology supported by the present framework.

1.1 Proximity Data and Stress Functions

Proximity data, the input to MDS, consist of dissimilarity information for *pairs of objects*. This contrasts with multivariate data that consist of attribute information for *individual objects*. If the objects are labeled $i = 1, ..., N$, we will assume that proximity data are given by dissimilarity values $D_{i,j}$. (If the data are given as similarities, a monotone decreasing transformation will convert them to dissimilarities.) The goal of MDS is to map the objects $i = 1, ..., N$ to "configuration" or "embedding" 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 $\|\mathbf{x}_i - \mathbf{x}_j\|$. 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.

The dissimilarity matrices of Figure 1 are simple examples with easily recognized error-free MDS solutions. The left hand matrix suggests mapping the five objects to an equispaced linear arrangement; the right hand matrix suggests mapping the three objects to a right triangle. The figure shows configurations actually found by MDS. The first configuration

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

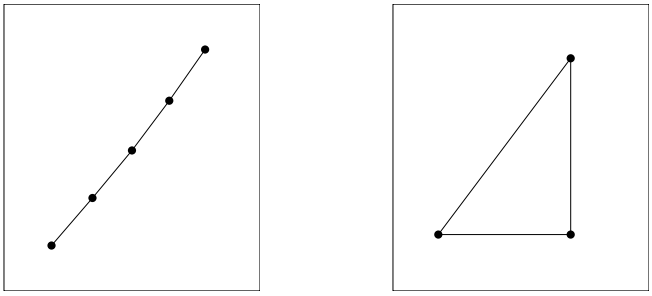

Figure 1: *Simple Examples of Dissimilarity Matrices and Their Optimal Scaling Solutions.*

can be embedded in $k = 1$ dimension, while the second needs $k = 2$ dimensions. 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. Kruskal defined MDS in terms of minimization of a loss function called "Stress", which is simply a measure of lack of fit between dissimilarities $D_{i,j}$ and fitted distances $\|\mathbf{x}_i - \mathbf{x}_j\|$. In the simplest case, Stress is a residual sum of squares:

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

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 configurations $(\mathbf{x}_1, ..., \mathbf{x}_N)^T$, thought of as $N \times k$-dimensional hypervectors of unknown parameters. The minimization can be carried out by straightforward gradient descent applied to Stress_D, viewed as a function on $\mathbb{R}^{Nk}$.

We note that MDS is blind to asymmetries in the dissimilarity data because

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

where C is an expression that does not depend on $\|\mathbf{x}_i - \mathbf{x}_j\|$. We therefore assume from now on that the dissimilarities are symmetrized. 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 several types of MDS, and they differ mostly in the loss 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**: In distance scaling dissimilarities are fitted by distances $\|\mathbf{x}_i - \mathbf{x}_j\|$ (Sections 1.1 and 4.1), whereas classical scaling transforms the dissimilarities D_{ij} to a form that is naturally fitted by inner products $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$. The transformation of dissimilarity data D_{ij} to "inner-product data" B_{ij} satisfies $D_{ij}^2 = B_{ii} - 2B_{ij} + B_{jj}$, thereby mimicking the corresponding identities for $\|\mathbf{x}_i - \mathbf{x}_j\|$ and $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ (Section 4.2).
- **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 dissimilarities simultaneously with the configuration.

We implemented both distance scaling and classical inner-product scaling, and both metric and nonmetric scaling. In all, four types of MDS are provided by crossing {metric, nonmetric} with {distance, classical}. The unusual case of nonmetric classical scaling is described in Section 4.2.

A 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 mean-centered configurations.

Another difference between classical and distance scaling is that distance scaling requires iterative minimization while classical scaling can be solved with inexpensive eigendecompositions. Just the same, we implemented classical scaling with iterative gradient descent on a loss function called "Strain", which parallels gradient descent on Stress in distance scaling. This computational uniformity has advantages because it is straightforward to introduce weights and missing values in Strain and Stress, which is not possible in eigendecompositions.

1.3 Applications of MDS

Here is an incomplete list of application areas of 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.

- A common use of MDS is 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 $\text{dist}(\mathbf{y}_i, \mathbf{y}_j) = D_{i,j}$, and use distance scaling to find lower-dimensional $\mathbf{x}_1, ..., \mathbf{x}_N \in \mathbb{R}^k$ ($k \ll K$) 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, whereas classical scaling is identical to principal components. (For an interpretation of multivariate analysis as distance approximation, see Meulman (1992).)
- In chemistry, MDS can be used for *molecular conformation*, that is, finding the spatial structure of molecules. This application differs from the above in that 1) actual distance information is available from experiments or theory, and 2) the only meaningful embedding dimension is $k = 3$. Configurations are here called "conformations." Some references are Crippen and Havel (1978), Havel (1991), Glunt et al. (1993), and Trosset (1998a). For an example, see Figure 10.
- A fourth use of MDS is for *graph layout*, an active area at the intersection of discrete mathematics and network visualization (Di Battista et al. 1999). Pioneers before their time were Kruskal and Seery (1980). From graphs one obtains dissimilarities by computing shortest-path lengths for all pairs of vertices. Resulting MDS configurations can be used to draw graphs with line segments indicating graph vertices, as in Figure 9.

We will show examples of data in all four categories.

1.4 Related Developments in Machine Learning

In recent years, the machine learning community developed an interest in nonlinear dimension reduction. The problem of flattening nonlinear manifolds led to localized approaches to dimension reduction that use small dissimilarities only, the intuition being that large dissimilarities reflect nonlinear warping in high-dimensional data space and should not be relied on. The following two approaches achieve such localization by preprocessing the dissimilarities and relying subsequently on plain classical MDS:

- In Isomap (Tenenbaum et al. 2000) one replaces large dissimilarities with shortest-path lengths computed from small dissimilarities. Intuitively, one replaces "cordal distances" with the lengths of shortest paths "within the manifold." Small dissimilarities may be defined as being among K nearest neighbors of either endpoints.
- In kernel PCA (principal components, Schölkopf et al. 1998) one creates inner-product data B_{ij} with "kernel tricks" such as $B_{ij} = \exp(-D_{ij}^2/(2\sigma^2))$. The point of this transformation is to create inner-product data B_{ij} that place two objects i and j about equally far from each other (namely, orthogonal to each other) whenever roughly $D_{ij} > 3\sigma$. Localization is achieved by choosing σ small.

In both cases the resulting proximity data are subjected to classical MDS, but in the present framework it is as easy to apply distance MDS as well as nonmetric versions of either that achieve better embeddings by estimating an additional transformation of the dissimilarities.

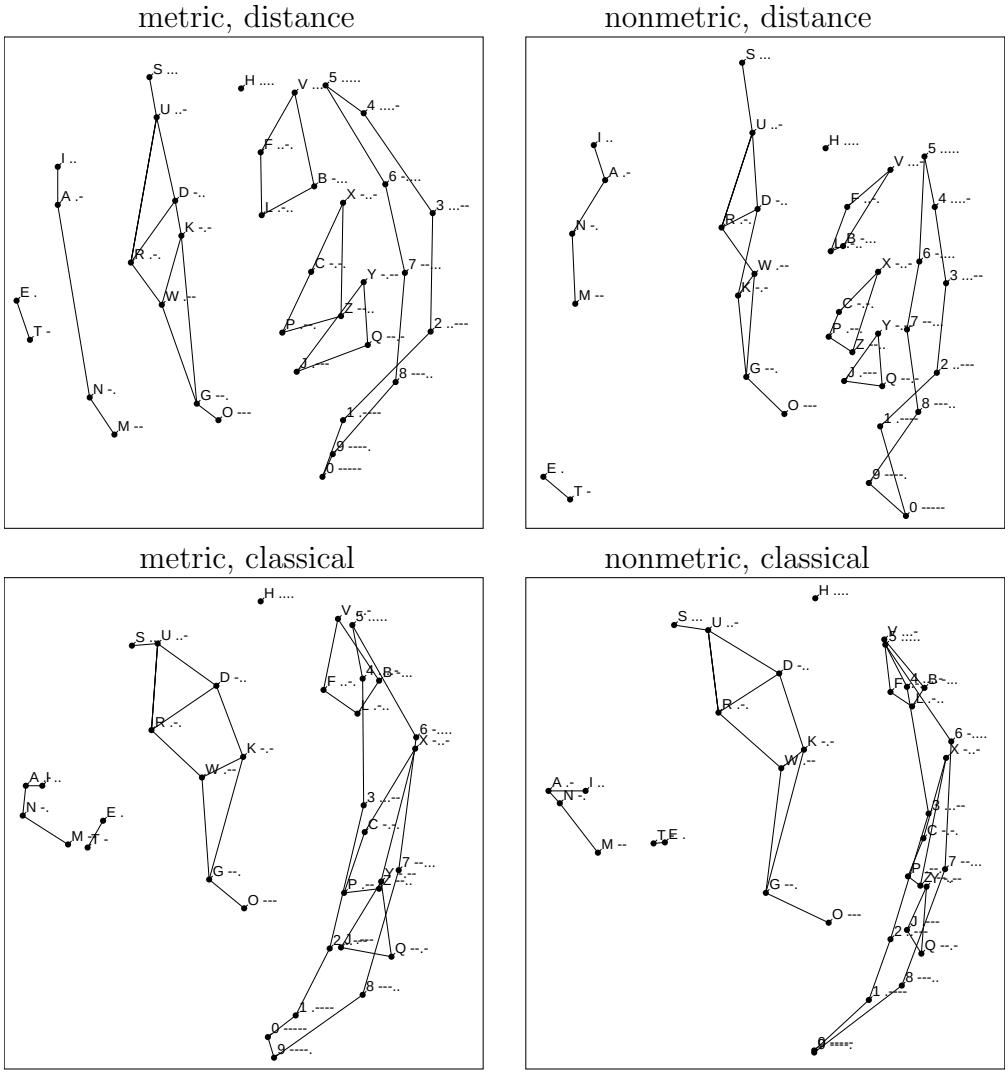

Figure 2: *Rothkopf's Morse Code Data: four 2-D configurations.*

A competing method, Local Linear Embedding (LLE, Roweis and Saul 2000) is conceptually unrelated to MDS. A more recent approach is Local MDS (LMDS, Chen and Buja 2007) which modifies the stress function by adding penalties that spread out and flatten manifolds.

1.5 A Classical Example: The Rothkopf Morse Code Data

For illustration we use as our running example the well-known Rothkopf Morse code data (1957), which are to MDS what Fisher's Iris data are to discriminant analysis. These data originated from an experiment in which inexperienced subjects were exposed to pairs of Morse codes in rapid order, and the subjects had to decide whether the codes in a pair were identical. The data were summarized in a table of confusion rates.

Confusion rates are similarity measures because two codes that are often confused are interpreted as "similar" or "close." For the conversion of similarities to dissimilarities one could in principle use any monotone decreasing transformation, but we used the following:

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

In other words, we interpreted the similarities as inner product data. 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). Unlike other conversion methods, this one has the desirable property $D_{i,i} = 0$. We also symmetrized the data (see Section 1.1).

Applying all four scaling methods in $k = 2$ dimensions 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 ("H"). 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 (for example: Shepard 1962; Kruskal and Wish 1978, p. 13; Borg and Groenen 1997, p. 59). This interpretation of the axes was achieved by rotating the configurations (which is permissible due to rotation invariance of the distances $\|\mathbf{x}_i - \mathbf{x}_j\|$ and inner products $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$). While we easily obtained interpretable axes with interactive rotations, one usually tries to achieve the same in factor analysis with automated rotation methods such as Varimax.

The four scaling methods exhibit significant differences. Nonmetric distance scaling produces probably the most satisfying configuration, with the exception of the placement of the codes of length 1 ("E" and "T"). Metric scaling suffers from circular bending but it places "E" and "T" in the most plausible location. The classical scaling methods 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 tends to achieve better compromises in lower dimensions.

2 Interactive MDS Operation

The principal use of MDS configurations is for visualization. Because configurations are essentially multivariate data, any MDS system calls for a multivariate data visualization tool. As some of us co-authored the XGobi and GGobi systems for data visualization, it was natural that we chose these as viewing engines (XGobi: Swayne, Cook and Buja 1998; Buja, Cook and Swayne 1996; GGobi: Swayne, Buja and Temple Lang 2003; Swayne, Temple Lang, Buja and Cook 2002). In effect, X/GGvis are master programs that create and feed X/GGobi. Figure 3 shows how this presents itself in the case of GGobi: a GGvis control panel for MDS (left), a GGobi window for viewing the configuration (center top), a GGobi control panel (center bottom), and a GGobi window for diagnostics (right). The basic sequence of MDS interactions is as follows:

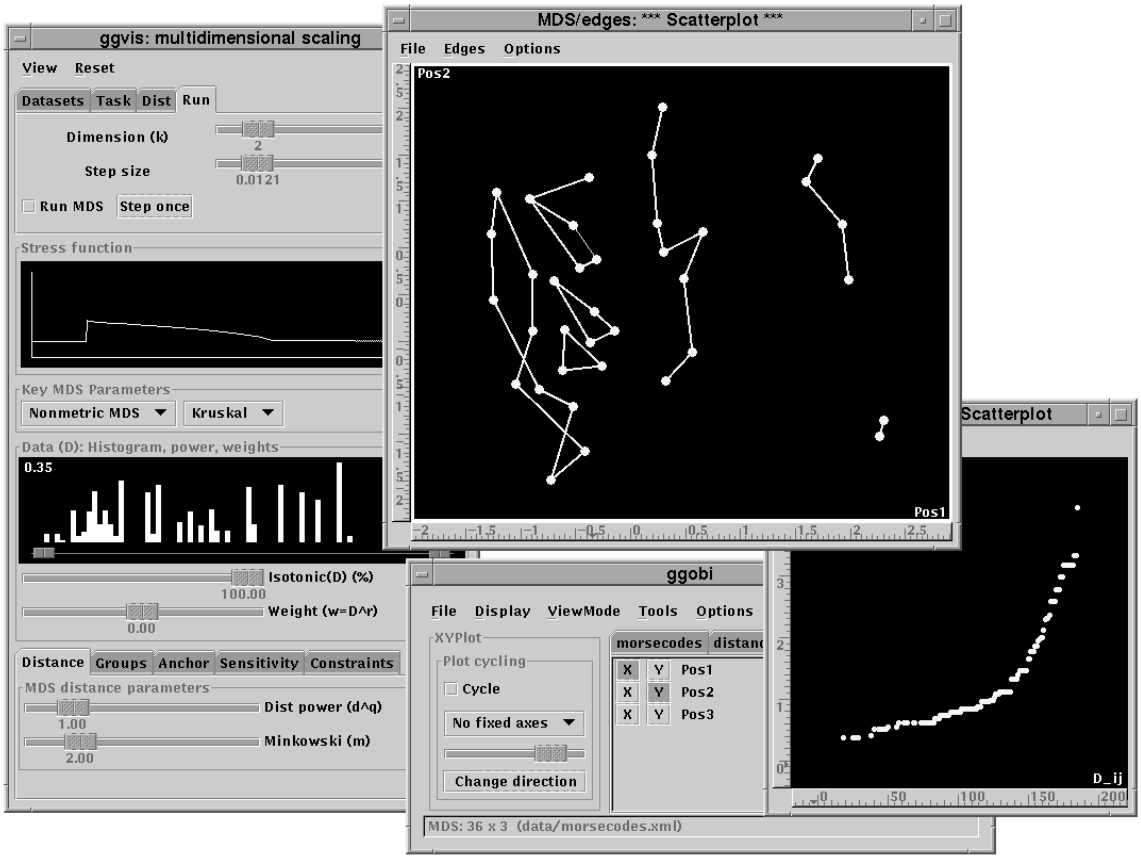

Figure 3: *The Major GGvis Windows. On the left is the master control panel, on the right is the GGobi window for the configuration. Below is the GGobi window for an optional Shepard plot.*

- Launch with dissimilarity data (an $N \times N$ matrix), multivariate data (an $N \times d$ matrix), or graph data (a list of integer pairs, possibly with weights) to perform, respectively, 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 (default: metric distance scaling).
- Choose a configuration dimension (default: 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. Finally, stop the optimization (toggle "Run MDS"). There is no automatic convergence criterion, and optimization does not stop on its own.
- Examine the shape of the optimized configuration: If $k = 2$, a plain X-Y scatterplot will do. If $k = 3$, use 3-D rotations, and if $k > 3$, use the grand tour.

- Interpret the configuration: Assuming informative object labels were provided with the input data, search the configuration by labeling points. If covariates are available, interpretation can be further aided by linked color or glyph brushing between covariate views and configuration views. As this is only a tentative search for interpretable structure, one should use transient (as opposed to persistent) brushing.
- Enhance the configuration: After acquiring a degree of familiarity with the configuration, use persistent brushing 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 and coloring the lines.
- Turn optimization back on and leave it continually running. Observe the response to
  - experimenting with various transformation parameters,
  - subsetting objects,
  - subsetting dissimilarities,
  - weighting dissimilarities,
  - manually moving points and groups of points, and
  - perturbing the configuration or restarting from random configurations.
- Stop optimization and perform diagnostics in a separate window that shows among other things a "Shepard plot" of the transformed dissimilarities and the fitted distances.

We described elsewhere (Swayne et al. 1998; Cook and Swayne 2007) operations such as 3-D rotations and grand tours, as well as (linked) brushing, labeling and line editing. Manually moving points is also part of the viewing engine (X/GGobi), but in conjunction with MDS optimization it takes on special importance and is therefore described in Section 3. MDS parameters as well as weighting and subsetting of dissimilarities affect the loss function and are therefore specific to MDS. They are the subject of Section 4.

3 Animated Optimization and Point Manipulation

Most software for MDS operates in a batch mode, even though many older programs have been wrapped in somewhat interactive PC environments. A pioneering system that is truly interactive is McFarlane and Young's "ViSta-MDS" (1994) whose development overlaps with first version of XGvis (Littman et al. 1992). Although the two systems were developed independently, they share two important capabilities:

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

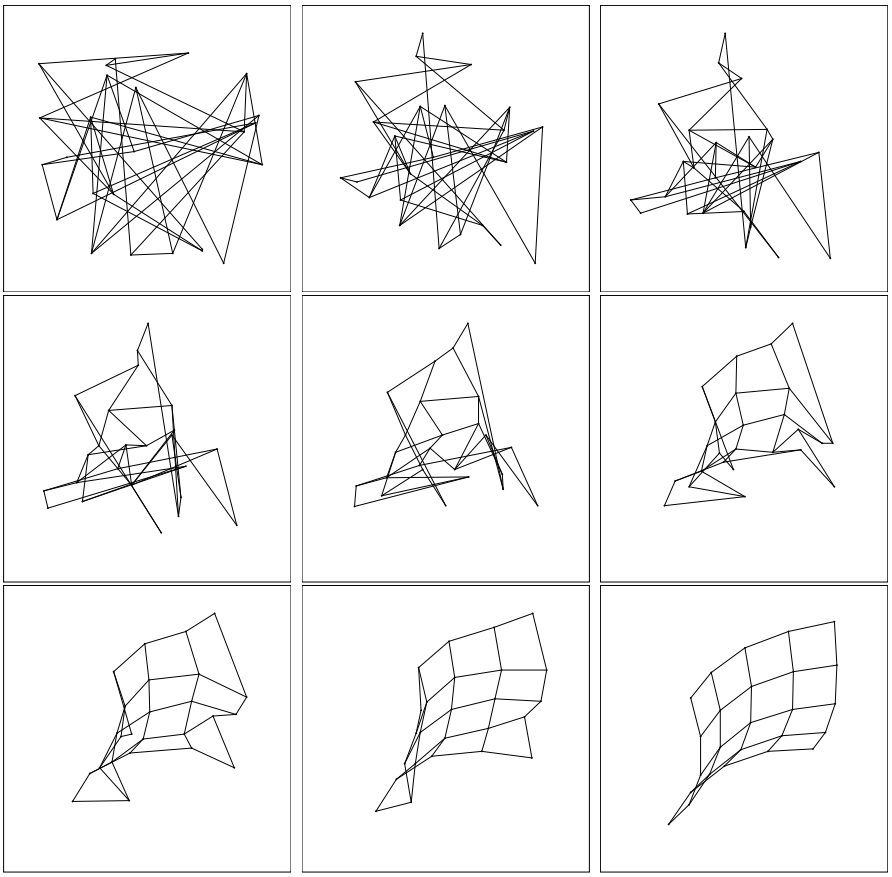

Figure 4: *Snapshots from a MDS Animation. The figure shows nine stages of a Stress minimization in three dimensions. It reconstructed a* 5×5 *square grid from a random configuration. The grid was defined as a graph with 25 nodes and 40 edges. The distances were computed as the lengths of the minimal paths in the graph (= city block-, Manhattan-, or* L_1*-distances). These distances are not Euclidean, causing curvature in the configuration.*

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 an *alternating mode of operation* in which users switch back and forth between animated optimization and manipulation. X/GGvis is by default in a *fluid mode of operation* in which the program runs in a never-ending optimization loop, with no stopping criterion whatsoever. The user can drag the configuration points while the optimization is in progress. The optimizer does not attempt to move the dragged point, but the other points "feel" the dislocation through the change in the loss function, and they are therefore slowly pulled because they try to position themselves in a local minimum configuration with regard to the dragged point. As soon as the dragged point is let go, it

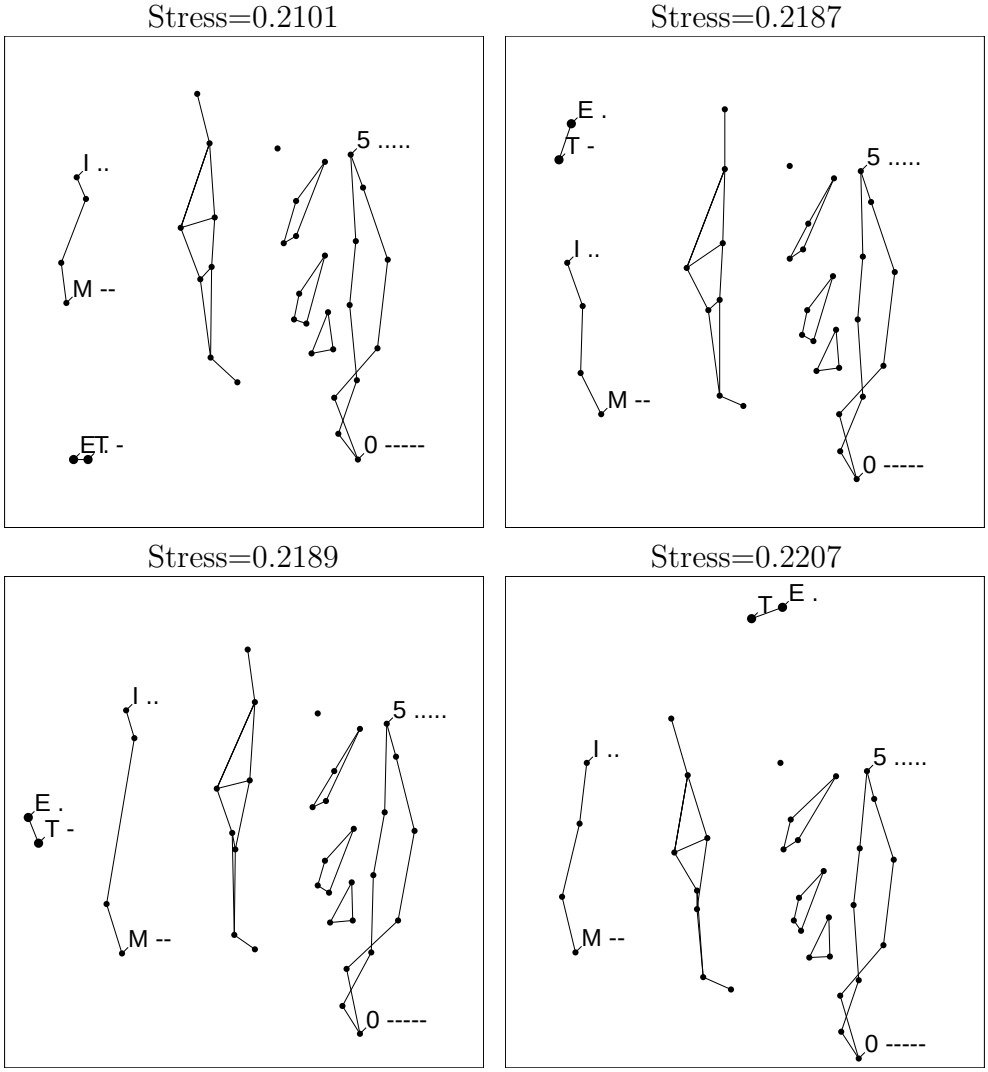

Figure 5: *Four Local Minima Found by Moving the Group* $\{E, T\}$ *into Different Locations.*

snaps into a position that turns the configuration into a local minimum of the loss function. The feel for the user is that of pricking and tearing the fabric of the configuration and thereby exploring deformations of its shape.

In addition to moving one point at a time, X/GGvis permits *moving groups of points* defined by 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 transformed by a third power (which mimics a nonmetric solution), we found 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* such as Gaussian random point clouds. 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 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.

4 Loss Functions: Stress and Strain

As mentioned, we use iterative minimization of loss functions even where eigendecompositions could be applied, the reason being that missing and weighted dissimilarities are handled with difficulty by the latter but trivially by the former. We now describe general forms of loss functions called Stress in distance MDS and Strain in classical MDS.

4.1 Stress

Although the simplest form of loss function for distance scaling is a residual sum of squares as in (1), it is customary to report Stress values that are standardized and unit-free. Stress may therefore take the form

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

This, however, is not our final form of Stress. A problem with residual sums of squares, which we gleaned from watching their optimization in real-time, is the effort spent on getting the size of the configuration right. This effort is entirely unnecessary because we are only interested in the shape of the configuration. Any two configurations $(\mathbf{x}_i)_i$ and $(t \cdot \mathbf{x}_i)_i$ that differ only by a size factor t are equivalent for practical purposes. It is therefore desirable to use a form of Stress that has essentially the same minimum configurations as (2) except that it is invariant under scale changes. For the particular case of residual sums of squares in linear models the solution to the problem is well-known: minimizing a residual sum of squares in a model with intercept is equivalent to maximizing a correlation up to a scale factor, which is what we are looking for. A general approach to making a loss function size invariant is by optimizing the size factor explicitly. In other words, a configuration is judged not by its actual value of the loss function, but by the value of the loss after scaling the configuration optimally. That is, we replace $\text{Stress}_D(\mathbf{x}_1, ..., \mathbf{x}_N)$ with $\min_t \text{Stress}_D(t \cdot \mathbf{x}_1, ..., t \cdot \mathbf{x}_N)$:

$$\min_t \text{Stress}_D(t \cdot \mathbf{x}_1, ..., t \cdot \mathbf{x}_N) \;=\; \left(1 \;-\; \frac{\left(\sum_{i,j} D_{i,j} \cdot \|\mathbf{x}_i - \mathbf{x}_j\|\right)^2}{\sum_{i,j} D_{i,j}^2 \cdot \sum_{i,j} \|\mathbf{x}_i - \mathbf{x}_j\|^2} \right)^{1/2} \tag{3}$$

The ratio inside the parentheses can be interpreted statistically as an uncentered squared correlation and geometrically as the squared cosine between the "vectors" $(D_{i,j})_{i,j}$ and $(\|\mathbf{x}_i - \mathbf{x}_j\|)_{i,j}$, and the complete right hand side is the sine between these two vectors. The size invariance of this form of Stress gives implementations the freedom to scale the configurations according to viewers' convenience.

However, (3) is still not the final form we use. Additional features we incorporate include 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.

The final form of Stress as implemented is shown below; details such as the summation set I and the weights $w_{i,j}$ are discussed in Subsections 4.4 and 4.5, respectively.

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

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

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

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

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

$\mathbf{x}_1, ..., \mathbf{x}_N \in \mathrm{I\!R}^k$, configuration points; $\;1 \leqslant k \leqslant 12$, default: $k = 3$

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

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

4.2 Strain, Metric and Nonmetric

Classical scaling is based on inner products which, unlike distances, depend on the origin. W.l.o.g. one can center configurations at the origin, $\sum_i \mathbf{x}_i = 0$, so their inner product matrices have zero marginal means, $\sum_k \langle \mathbf{x}_k, \mathbf{x}_j \rangle = \sum_k \langle \mathbf{x}_i, \mathbf{x}_k \rangle = 0$. It is now easy to derive a transformation of distance data D_{ij} to inner-product data B_{ij} with a heuristic:

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

The reasoning is as follows: If $\|\mathbf{x}_i - \mathbf{x}_j\|^2$ approximates D_{ij}^2, then $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ should approximate

$$\tilde{D}_{i,j} := -D_{ij}^2/2 \tag{4}$$

after removing the marginal means, a process known as "double-centering":

$$B_{ij} = \tilde{D}_{ij} - \tilde{D}_{i\bullet} - \tilde{D}_{\bullet j} + \tilde{D}_{\bullet\bullet} \;, \tag{5}$$

where $\tilde{D}_{i\bullet}$, $\tilde{D}_{\bullet j}$ and $\tilde{D}_{\bullet\bullet}$ are row, column and grand means of $\tilde{D}$.

We call "Strain" any loss function that measures the lack of fit between inner products $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ and inner-product data B_{ij}. A simple form of Strain analogous to the form (2) of Stress is a standardized residual sum of squares:

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

As with Stress, it is desirable and feasible to create a size invariant version of Strain by minimizing over a scale factor which introduces a ratio that can be interpreted as an uncentered correlation or a cosine:

$$\min_t \; \text{Strain}_D(t \cdot \mathbf{x}_1, ..., t \cdot \mathbf{x}_N) \;=\; \left(1 \;-\; \frac{\left(\sum_{i,j} B_{i,j} \cdot \langle \mathbf{x}_i, \mathbf{x}_j \rangle \right)^2}{\sum_{i,j} B_{i,j}^2 \cdot \sum_{i,j} \langle \mathbf{x}_i, \mathbf{x}_j \rangle^2} \right)^{1/2} \tag{6}$$

Again, this is not the final form of Strain we use. The most involved addition to Strain is a nonmetric feature for the estimation of a monotone transformation of the dissimilarities. Interpreting Strain as a function of the dissimilarities D_{ij} by way of (4) and (5), the problem is to transform the values $D_{i,j}$ in such a way that a better Strain is obtained. A solution was proposed by Trosset (1998b), and we give here an alternative derivation with simplifications that permit us to fit the nonmetric Strain minimization problem into the existing framework. Classical nonmetric scaling is nonstandard, but it fills a conceptual gap by permitting us to cross {metric, nonmetric} with {classical scaling, distance scaling}. The properties of nonmetric classical scaling are not well-understood, but the implementation in X/GGvis will hopefully remedy the situation. We start with the following observations:

Lemma: *If B is constructed as above with double centering of $\tilde{D}$, any configuration that minimizes the Strain* (6) *is centered at the origin. If, on the other hand, configurations are*

constrained to be centered at the origin, replacing B with $\tilde{D}$ in (6) *preserves the minimizing configurations.*

Proof: If A and C are $N \times N$ matrices, denote by $\langle A, C\rangle_F = \text{trace}(A^T C) = \sum_{i,j} A_{i,j} C_{i,j}$ the Frobenius inner product, and by $\|A\|_F^2 = \langle A, A\rangle$ the squared Frobenius norm. Furthermore, let $\mathbf{e} = (1, ..., 1)^T \in \mathbb{R}^N$ and I be the $N \times N$ identity matrix, so that $P = I - \mathbf{e}\mathbf{e}^T/N^{1/2}$ 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 i'th row is $\mathbf{x}_i^T$, 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 $PX = X$. Using repeatedly $\langle PA, C\rangle_F = \langle A, PC\rangle_F$ one derives

$$\langle B, XX^T\rangle_F \;=\; \langle P\tilde{D}P, XX^T\rangle_F \;=\; \langle \tilde{D}, P(XX^T)P\rangle_F \;=\; \langle \tilde{D}, (PX)(PX)^T\rangle_F \;,$$

which proves that the numerator in (6) does not change when B is replaced with the uncentered matrix $\tilde{D}$ but X is replaced with the centered matrix PX. As for the denominator term $\sum_{ij}\langle \mathbf{x}_i, \mathbf{x}_j\rangle^2 = \|XX^T\|_F^2$, using $\langle PA, (I-P)C\rangle_F = 0$ one obtains the decomposition

$$\|XX^T\|_F^2 \;=\; \|PXX^TP\|_F^2 \;+\; \|(I-P)XX^T(I-P)\|_F^2 \;,$$

which proves that the Strain decreases when an uncentered X is replaced with a centered PX. Lastly, replacing $\sum_{ij} B_{ij}^2 = \|B\|_F^2$ with $\sum_{ij} \tilde{D}_{ij}^2 = \|\tilde{D}\|_F^2$ in the denominator of (6) changes the value of the criterion but it does not affect the minimum Strain configurations. □

The modification of the Strain (6) suggested by the Lemma is therefore:

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

under the constraint $\sum_i \mathbf{x}_i = 0$. In light of $\tilde{D}_{ij} = -D_{ij}^2/2$ being a descending transformation of D_{ij}, it is natural for a nonmetric approach to classical scaling to let the Strain function determine a transformation $f(-D_{ij})$. A nonmetric form of Strain is therefore:

$$\text{Strain}_D(\mathbf{x}_1, ..., \mathbf{x}_N) \;=\; \left(1 \;-\; \frac{\left(\sum_{i,j} f(-D_{i,j}) \cdot \langle \mathbf{x}_i, \mathbf{x}_j\rangle\right)^2}{\sum_{i,j} f(-D_{i,j})^2 \cdot \sum_{i,j}\langle \mathbf{x}_i, \mathbf{x}_j\rangle^2} \right)^{1/2} . \tag{7}$$

Similar to nonmetric distance scaling, the monotone ascending transformation $f()$ of $-D_{ij}$ can be estimated with isotonic regression of the response $(\langle \mathbf{x}_i, \mathbf{x}_j\rangle)_{ij}$ on the predictor $(-D_{ij})_{ij}$.

We have arrived at the nearly final form of Strain. As with Stress, the Strain we actually use includes power transformations of dissimilarities in metric mode as well as weighting and omitting of dissimilarities. The final form of Strain as implemented is shown below; again, details such as the summation set I and the weights $w_{i,j}$ are discussed in Subsections 4.4 and 4.5, respectively.

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

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

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

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

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

$\mathbf{x}_1, ..., \mathbf{x}_N \in \mathbb{R}^k,$ configuration points, constrained to $\sum \mathbf{x}_i = 0$
$1 \leqslant k \leqslant 12,$ default: $k = 3$

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

4.3 Parameters of Stress and Strain

Stress and Strain have parameters that are easily subjected to interactive user control:

- 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 reduced or configuration dimension k. The conventional choice is $k = 2$, but with 3-D rotations available it is natural to chose $k = 3$ as the default. Recall that for $k \geqslant 4$ viewing is possible with grand tours.
- Both metric and nonmetric scaling permit transformations of the dissimilarities that are controlled by certain parameters:
  - *Metric scaling* uses a power transformation with a user-controlled exponent p: D_{ij}^p and $-D_{ij}^{2p}$ for distance and classical scaling, respectively (default: $p = 1$). In our experience powers as high as 4 have proven useful. An interesting power is 0: it describes objects that are all equally far from each other and hence form a simplex. This is the "null case" of total indiscrimination.

– *Nonmetric scaling* is defined by the use of isotonic regression to nonparametrically estimate a transformation of the dissimilarities. The point of nonmetric approaches is to use only the ranks of the dissimilarities D_{ij}, not their actual values. We found it useful to allow convex mixing of the isotonic transformation with the nonmetric power transformation. The mixing 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 nonmetric and metric solutions. Moving s temporarily below 1 can help a nonmetric configuration recover when it gets trapped in a degeneracy (usually clumping of the configuration points in a few locations with near-zero Stress value and an isotonic transformation that degenerated to a step function; see Borg and Groenen (1997), Sections 13.2-3).

- 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 $p = q = 2$. SStress fits squared distances to squared dissimilarities, and it is used in the influential MDS software "ALSCAL" by Takane et al. (1977). In our experience SStress is somewhat less robust than Stress, as the former is even more strongly influenced by large dissimilarities than Stress.
- There is finally a choice of the distance measure in configuration space for distance scaling. We use Minkowski (also called Lebesgue) metrics other than Euclidean and subject the Minkowski parameter m to user manipulation. The default is Euclidean, $m = 2$; the city block or L_1 metric is the limiting case $m \to 1$.

4.4 Subsetting

The loss functions can trivially handle missing values in the dissimilarity matrix: pairs (i, j) for which D_{ij} is missing are simply dropped from the summations. Through the deliberate use of missing values one can implement certain extensions of MDS such as multidimensional unfolding (see Borg and Groenen (1997), chapter 14, in particular their Figure 14.1). Missing dissimilarities can be introduced in various ways: They can be NA coded in the input file, or they can be introduced interactively through user-generated conditions that include thresholding of dissimilarities, random selection with probability α, and selection based on existing groups marked by colors or glyphs. Here is a symbolic description of the summation set of both Stress and Strain, followed by a detailed discussion:

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

Thresholding: Threshold parameters T_0 and T_1 for the conditions $T_0 \leqslant D_{i,j} \leqslant T_1$ can be used to check the influence of large and small dissimilarities. 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 in turn the largest third and the smallest third of the dissimilarities (T_1 = upper tercile and T_0 = lower tercile, respectively). They found devastating effects when removing the largest third, but benign effects when removing the smallest third. With user-chosen thresholding the degree to which this behavior holds can be explored for every dataset individually.

Random selection of dissimilarities with a user-selected probability α is another way to investigate the stability of configurations. By repeatedly triggering this feature while optimization is continuously running, users obtain 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.

Group-based selection of dissimilarities constitutes an extremely powerful tool for MDS. Groups of objects can be defined in input files by color and/or glyph attributes, or they can be generated interactively with brushing operations on configuration points. The following are the operations we implemented:

- **Subsetting objects:** Remove some objects and scale the remaining objects. Removal is achieved by simply hiding groups. It is a common experience that subsets often reveal new structure when freed from constraints imposed by the whole dataset. For example, the circular structure of the digits in the Morse code data becomes fully visible only when the digits are scaled separately (Buja and Swayne 2002).
- **Within-groups scaling:** Remove the dissimilarities that cross groups. This option can be useful no only for finding but also comparing group-internal structure. We therefore found this option somewhat more powerful than the previous one. — 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. — We 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. Between-groups scaling with two groups is called multidimensional unfolding (see, for example, Borg and Groenen 1997, chapter 14). The case of only two groups is the 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. Therefore, floaters do not affect each other, and 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, hence configurations for the anchors can 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 6.2).

In practice we usually start with scaled anchors and later switch to fixed anchors. Then, while the optimization is running, we drag the anchor points into new locations 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). Anchored scaling is also reminiscent of so-called Tutte scaling in graph drawing; for an application in a multivariate analysis context, see De Leeuw and Michailidis (2000).

4.5 Weights

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

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

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

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 loss 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 is a gradual form of moving between conventional scaling, within-groups scaling, and between-groups scaling. The latter are our ideas, while the weight-based gradual version is due to Priebe and Trosset (personal communication).

Weighting is computationally more costly than subsetting. The latter saves time because some dissimilarities do not need to be looked at, while weights are costly in terms of memory because they are stored to save power operations in each iteration.

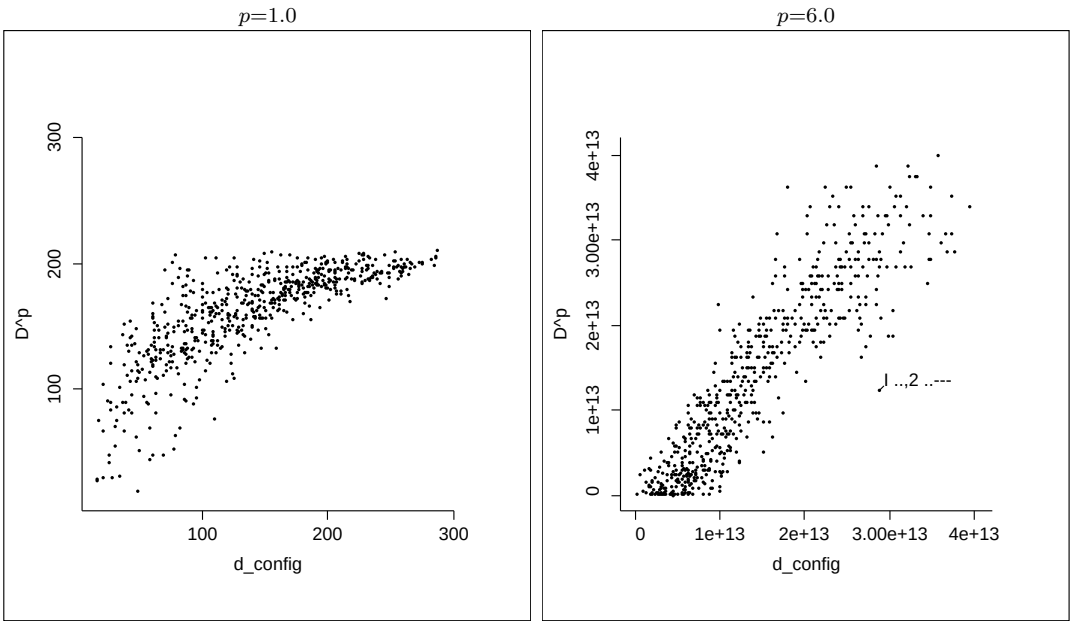

Figure 6: *Two Diagnostic Plots for Configurations from Metric Distance Scaling of the Morse Code Data. Left: raw data, power* $p = 1$*; right: transformed data, power* $p = 6$*. An outlier is marked in the right hand plot: the pair of codes* $(\mathrm{I}, 2) \sim (\cdot\,\cdot, \cdot\,\cdot - - - -)$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

5 Diagnostics

A standard diagnostic in MDS is the Shepard plot, that 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.

Diagnostics such as the Shepard plot are of a very different nature from the display of the configuration. For one thing they may be much more expensive because they may show $\binom{N}{2}$ points for N configuration points. They therefore call for a separate viewing engine (X/GGobi window) that is not continuously updated but represents a fixed snapshot at a given time. We found it useful to expand on the diagnostic variables and provide the following whenever diagnostics are generated:

- the dissimilarities: $D_{i,j}$;
- the fitted quantities: $d_{i,j} = \|\mathbf{x}_i - \mathbf{x}_j\|$ for distance scaling and $b_{i,j} = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$ for classical scaling;
- the transformation of the dissimilarities: $f(D_{i,j})$ for distance scaling and $f(-D_{i,j})$ for classical scaling, which is a power for metric scaling and an isotonic transformation for nonmetric scaling;

- the residuals: $r_{i,j} = f(D_{i,j}) - d_{i,j}$;
- the weights: $w_{i,j}$, which may be a power of the dissimilarities;
- the row index: i;
- the column index: j.

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 goodness of fit. An example is shown in Figure 6.

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 MDS modes in which dissimilarities have been removed from the Stress or Strain or if some dissimilarities are otherwise 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 loss function.

If the objects are given labels, 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 Optimization and the Fluid Mode of Operation

For minimization of Stress and Strain we mostly use Kruskal's (1964b) gradient descent method. This choice was based on the simplicity with which gradient methods generalize to loss functions with arbitrary weights and non-Euclidean Minkowski metrics. An alternative, not currently implemented, would be the popular SMACOF algorithm (Borg and Groenen 1997, and references therein). For metric scaling, Kruskal's method is plain gradient descent on the loss 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 we submit the stepsize to interactive user control. It is scaled in relation to the size of the configuration such that, 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 of the N row vectors from their respective mean vectors in $\mathbb{R}^k$.

As indicated earlier (Section 3), the main reason for not providing an automated stopping criterion is to enable a fluid mode of operation whereby users continously manipulate parameters and configurations and receive instantaneous feedback from the ongoing optimization. Another reason for the absence of a stopping criterion is that non-trivial local movement in

the configuration may be visually apparent even when the descent in the loss function has become negligible. User decision based on visual feedback provides a superior method of stopping that avoids premature as well as wasteful optimization.

6.2 Gradient Descent on the Configuration and Force Graphs

We describe details of the gradient descent step with a fixed transformation. The goal is to bring conventional MDS and anchored MDS (Section 4.4) into a single computational framework. The discussion is somewhat arcane but there is reward in the achieved unification. 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 form

$$\frac{Num}{Denom} \;=\; \frac{\sum_{(i,j)\in I}\; w_{i,j}\cdot g(D_{i,j})\cdot s(\mathbf{x}_i,\mathbf{x}_j)}{\left(\sum_{(i,j)\in I}\; w_{i,j}\cdot s(\mathbf{x}_i,\mathbf{x}_j)^2\right)^{1/2}}\,, \tag{8}$$

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. All terms are (or are assumed) symmetric in i and 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 we determine the size of gradient steps as a fraction of the size of the configuration, we only need the partial gradients up to a constant factor:

$$\frac{\partial}{\partial \mathbf{x}_i}\frac{Num}{Denom} \;\propto\; \sum_{j\in\{j|(i,j)\in I\}} w_{i,j}\left(g(D_{i,j}) - \frac{Num}{Denom^2}\; s(\mathbf{x}_i,\mathbf{x}_j)\right)\frac{\partial}{\partial \mathbf{x}}\bigg|_{\mathbf{X}=\mathbf{X}_i} s(\mathbf{x},\mathbf{x}_j)\;, \tag{9}$$

In the derivation of this formula we used symmetry of all terms in i and j and also symmetry of the set I: $(i,j)\in I \Rightarrow (j,i)\in I$. The summation should really extend over the set $\{j|\,(i,j)\in I \text{ or } (j,i)\in I\,\}$, but if I is symmetric it is sufficient to sum over the reduced set $\{j\,|\,(i,j)\in I\}$. Reduced summation lends an intuitive interpretation to partial gradients: The summand indexed by j is the contribution of $\mathbf{x}_j$ to the gradient movement of the point $\mathbf{x}_i$. As such, it can be interpreted as the "force" exerted by $\mathbf{x}_j$ on $\mathbf{x}_i$. The reduced summation means that under the symmetry assumptions it is sufficient to consider the force exerted by $D_{i,j}$ on $\mathbf{x}_i$ only, although strictly speaking both $D_{i,j}$ and $D_{j,i}$ exert force.

We now carry reduced summation over to certain types of non-symmetric sets I, and we start with a simple illustration. Consider the case where the first column is missing: $D_{i,1} = NA\;\forall i$ or, equivalently, $I = \{(i,j)\,|\,1\leqslant i\leqslant N,\; 2\leqslant j\leqslant N\}$. The missing first column means that $\mathbf{x}_1$ does not contribute to the partial gradient (9) of any point whatsoever, but its own 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 $\{j \mid D_{1,j} \neq NA\} = \{j \mid 2 \leqslant j \leqslant N\}$. This example shows that a suitable NA pattern in the dissimilarities permits us to implement anchored scaling in addition to conventional scaling, but reduced summation in combination with general summation sets I is of much greater generality as the following discussion of the group-based scaling methods of Section 4.4 shows:

- Within-groups scaling, illustrated with two groups:

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

  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, illustrated with two groups (multidimensional unfolding):

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

  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 anchors (a form of external unfolding):

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

  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 anchors (another form of external unfolding):

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

  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 and the corresponding summation sets I 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

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

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, multi-partite graph, that is, the node set is decomposed into two or more disjoint partitions, and the edge set is the set of edges from any one partition to any 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.

6.3 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 X/GGvis 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 X/GGvis was $N = 3648$, but larger N are feasible with more patience.

Among the four types of MDS, the nonmetric varieties are not 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. It should be kept in mind, however, that our implementation of classical scaling minimizes Strain, which is more flexible but also more costly than eigendecompositions that could be used if one assumed no nonmetric option, no missing dissimilarities, and no weights.

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. Our implementations do not allocate a weight matrix 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.

7 Examples of MDS Applications

We give some data examples that demonstrate the wide applicability of MDS and the usefulness of X/GGvis.

- **Dissimilarity data from computer usage:** As part of a project on intrusion detection at AT&T Labs (Theus and Schonlau 1998), users were characterized by their logs of UNIX®operating system commands that they typed during a number of work days. From each log, pairs of commands were characterized by a dissimilarity value that measured how far spaced the commands were on average (for example, there might be 3.2 other commands between the commands "spell" and "eqn" on average). Each dissimilarity matrix was considered as the signature of a user. MDS was used to create 2-D maps of the commands for a particular user. Examples from two users are shown in Figure 7. One user was a member of technical staff, the other an administrative assistant. The maps differ in characteristic ways, and they have very intuitive meanings in terms of general activities such as start up at the beginning of the day, followed by clusters of programming and data analysis activities by the member of technical staff, and by clusters of e-mailing and word processing by the administrative assistant.

- **Dimension reduction:** From a multivariate dataset with eight demographic and telephone usage variables for 1926 households we computed a Euclidean distance matrix after standardizing the variables. Distance MDS was used to reduce the dimension from eight to two by creating a 2-D configuration. Figure 8 shows the result, side by side with a 2-D principal component projection (equivalent to classical MDS in this case). While distance scaling squeezes 20% more variance into two dimensions than PCA, its map shows rounding on the periphery that may be artifactual. On the other hand, MDS seems to be better than PCA at spreading out four market segments (marked by glyphs) that were found with k-means. This example indicates that comparison of methods on the same dataset may be useful for detecting trade-offs.

- **Social networks of phone callers:** The development of XGvis was originally motivated by the problem of laying out graphs and in particular social networks such as telephone call graphs in more than two dimensions and using XGobi as a display device (Littman et al. 1992). In order to lay out a graph with MDS, one enters it as a list of object pairs and the software computes a dissimilarity matrix as a "shortest-path metric" (the smallest number of steps from one vertex to another within the graph). All configurations are then shown with line segments between points that are connected by an edge. Figure 9 shows an example of a call graph with 110 nodes layed out in 2-D by both classical and distance scaling. It is obvious that distance scaling succeeds in spreading out the vertices while classical scaling collapses some due to the integer nature of the dissimilarities.

UNIX is a registered trademark of The Open Group.

- **Molecular Conformation:** This is the task of embedding molecules in 3-D based on partial information about the chemical bonds. We give an example that amounts to a graph layout: D. Asimov (1998) devised a method for describing all possible capped nanotubes. His construction produces graphs with carbon atoms as nodes and bonds as edges. We show the graph layout generated with distance scaling of one such nanotube in Figure 10. Although the shortest-path metric of this molecular graph is far from Euclidean, the embedding reflects the intrinsic curvature quite well and certainly makes an excellent start for a more specialized chemical bond optimizer.

Software availability

The XGvis and GGvis systems can be freely downloaded from:

www.research.att.com/areas/stat/xgobi/
www.ggobi.org

GGvis is more recent and programmable from other languages such as R, Perl and Python.

Acknowledgments

We thank Jon Kettenring and Daryl Pregibon who supported this work when they were managers at Bellcore (now Telcordia) and AT&T Labs, respectively. We are indebted to Brian Ripley for his port of XGvis/XGobi to Microsoft Windows^{TM}. We finally thank an associate editor and a reviewer for extensive comments and help with scholarship.

References

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

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

[3] Buja, A., Cook, D., and Swayne, D. F. (1996), "Interactive high-dimensional data visualization," *J. of Computational and Graphical Statistics*, **5**, pp. 78–99.

[4] Buja, A., and Swayne, D. F. (2002), "Visualization Methodology for Multidimensional Scaling," *J. of Classification*, **19**, pp. 7-43.

[5] Chen, L., and Buja, A. (2007), "Local Multidimensional Scaling for Nonlinear Dimension Reduction, Graph Layout and Proximity Analysis," (under review, preprint at www-stat.wharton.upenn.edu/~buja/PAPERS/lmds-chen-buja.pdf).

Microsoft Windows is a trademark of Microsoft, Inc.

[6] Cox, R. F., and Cox, M. A. A. (1994), *Multidimensional Scaling*, London: Chapman & Hall.

[7] Cook, D., and Swayne, D. F. (2007), "Interactive and Dynamic Graphics for data Analysis," New York: Springer Verlag.

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

[9] De Leeuw, J., and Michailidis, G. (2000), "Graph-layout techniques and multidimensional data analysis," in Papers in Honor of T.S. Ferguson, Le Cam, L. and Bruss, F.T. (eds), IMS Monograph Series (Hayward, CA), pp. 219-248.

[10] Di Battista, G., Eades, P., Tamassia, R., and Tollis, I. G. (1999), *Graph Drawing: Algorithms for the Visualization of Graphs*, Indianapolis: Prentice Hall.

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

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

[13] 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.

[14] Kruskal, J. B., and Wish, M. (1978), *Multidimensional Scaling*, Beverly Hills and London: Sage Publications.

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

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

[17] 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.

[18] Littman, M., Swayne, D. F., Dean, N., and Buja, A. (1992), "Visualizing the embedding of objects in Euclidean space," *Computing Science and Statistics*, **24**, pp. 208-217.

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

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

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

[22] Roweis S. T., and Saul, L. K. (2000), "Nonlinear dimensionality reduction by local linear embedding," *Science*, **290**, 2323-2326.

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

[24] Schölkopf, B., Smola, A. J., and Müller, K.-R. (1998). "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, **10**, pp. 1299-1319.

[25] 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.

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

[27] Swayne, D.F., Buja, A., Temple-Lang, D. (2003), "Exploratory Visual Analysis of Graphs in GGobi," proceedings of the *Third Annual Workshop on Distributed Statistical Computing* (DSC 2003), Vienna.

[28] Swayne, D.F., Temple-Lang, D., Buja, A., and Cook, D. (2002), "GGobi: Evolving from XGobi into an Extensible Framework for Interactive Data Visualization," *Journal of Computational Statistics and Data Analysis*.

[29] 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.

[30] Tenenbaum, J. B., Silva, V. d., and Langford, J. C. (2000), "A global geometric framework for nonlinear dimensionality reduction," *Science*, **290**, 2319-2323.

[31] Theus, M. and Schonlau, M., (1998), "Intrusion Detection Based on Structural Zeroes," *Statistical Computing & Graphics Newsletter*, **9**, pp 12-17. Alexandria, VA: American Statistical Association.

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

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

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

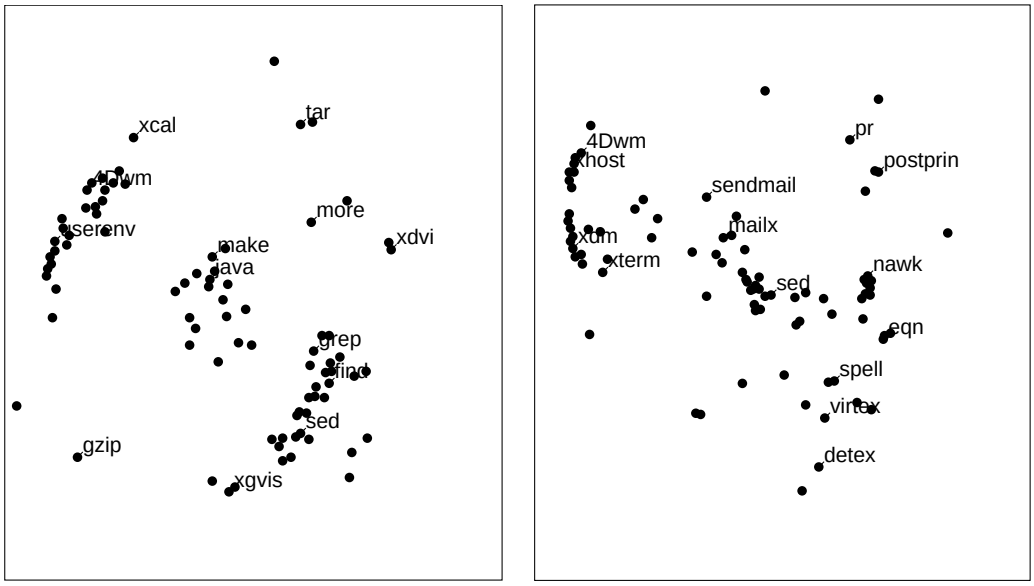

Figure 7: *Maps of Computer Commands for Two Individuals.*
Left: a member of technical staff who programs and manipulates data (Stress=0.29).
Right: an administrative assistant who does e-mail and word processing (Stress=0.34).
The labels show selected operating system commands used by these individuals.

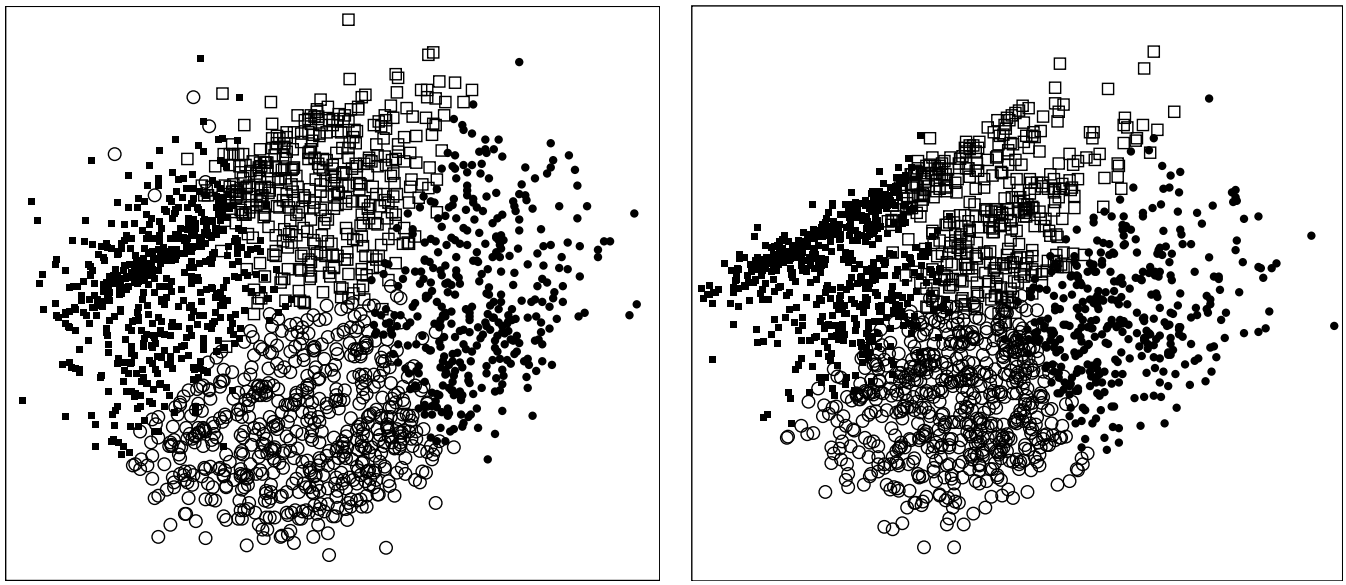

Figure 8: *Marketing Segmentation Data. Left: MDS reduction to 2-D; right: largest two principal components. The glyphs represent four market segments constructed with k-means clustering using four means.*

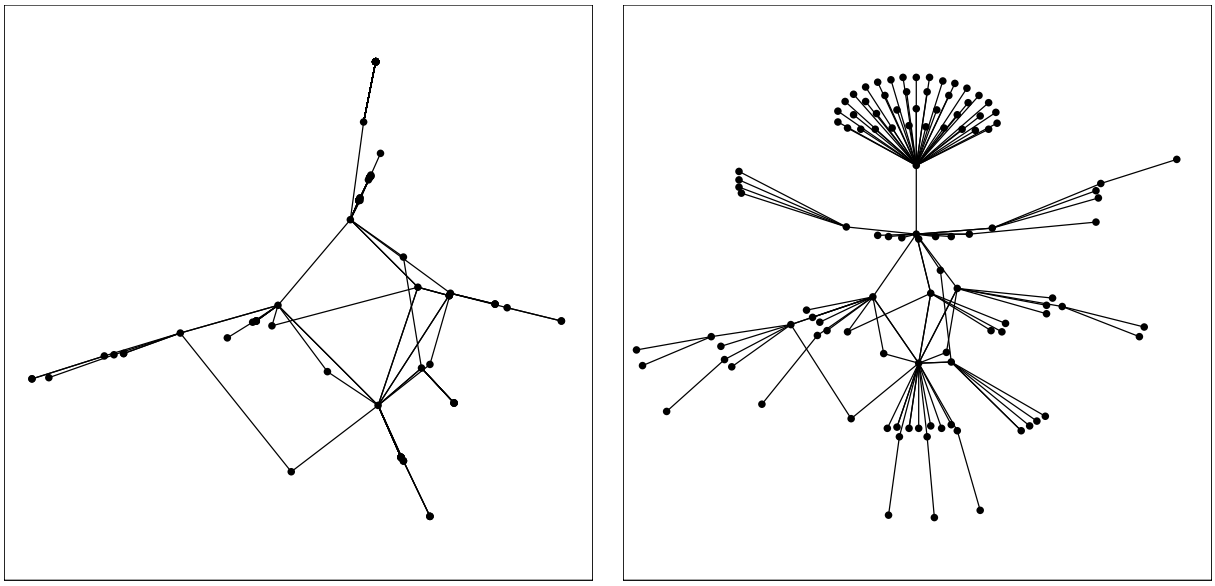

Figure 9: *A Telephone Call Graph, Layed Out in 2-D. Left: classical scaling (Stress=0.34); right: distance scaling (Stress=0.23). The nodes represent telephone numbers, the edges represent the existence of a call between two telephone numbers in a given time period.*

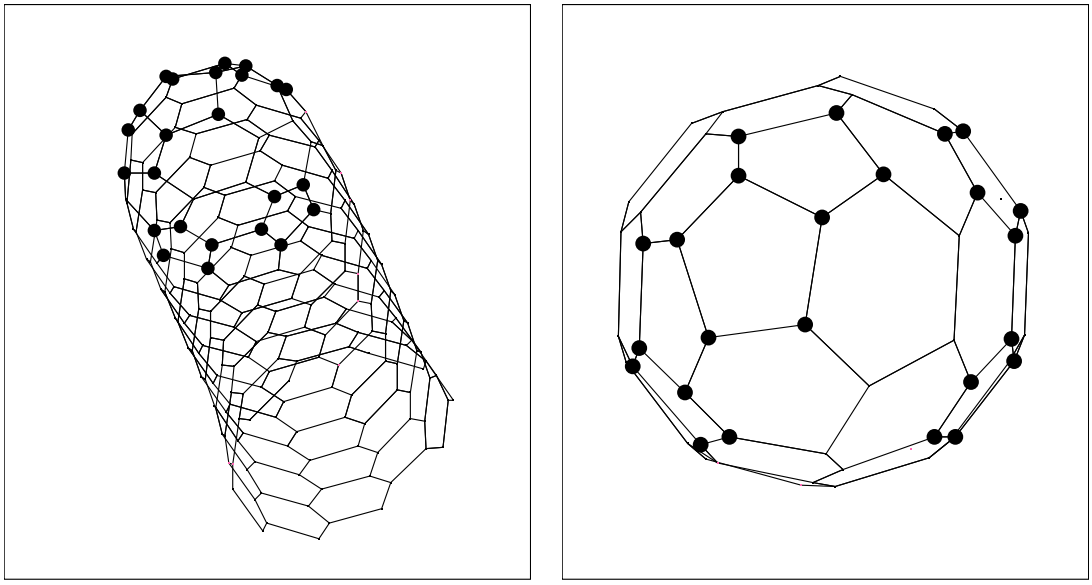

Figure 10: *Nanotube Embedding. One of Asimov's graphs for a nanotube is rendered with MDS in 3-D (Stress=0.06). The nodes represent carbon atoms, the lines represent chemical bonds. The right hand frame shows the cap of the tube only. The highlighted points show some of the pentagons that are necessary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