Journal of Machine Learning Research 5 (2004) 399–420j Submitted 7/03; Revised 1/04; Published 4/04

Distributional Scaling: An Algorithm for Structure-Preserving Embedding of Metric and Nonmetric Spaces

Michael Quist MJQ1@CORNELL.EDU
Department of Chemistry and Biochemistry
University of California at Los Angeles
Los Angeles, CA 90095, USA

Golan Yona GOLAN@CS.CORNELL.EDU
Department of Computer Science
Cornell University
Ithaca, NY 14853, USA

Editor: Bin Yu

Abstract

We present a novel approach for embedding general metric and nonmetric spaces into low-dimensional Euclidean spaces. As opposed to traditional multidimensional scaling techniques, which minimize the distortion of pairwise distances, our embedding algorithm seeks a low-dimensional representation of the data that preserves the structure (geometry) of the original data. The algorithm uses a hybrid criterion function that combines the pairwise distortion with what we call the geometric distortion. To assess the geometric distortion, we explore functions that reflect geometric properties. Our approach is different from the Isomap and LLE algorithms in that the discrepancy in distributional information is used to guide the embedding. We use clustering algorithms in conjunction with our embedding algorithm to direct the embedding process and improve its convergence properties.

We test our method on metric and nonmetric data sets, and in the presence of noise. We demonstrate that our method preserves the structural properties of embedded data better than traditional MDS, and that its performance is robust with respect to clustering errors in the original data. Other results of the paper include accelerated algorithms for optimizing the standard MDS objective functions, and two methods for finding the most appropriate dimension in which to embed a given set of data.

Keywords: Embedding, multidimensional scaling, PCA, earth-mover's distance

1. Introduction

Embedding is concerned with mapping a given space into another space, often Euclidean, in order to study the properties of the original space. This can be especially effective when the original space is a set of abstract objects (e.g., strings, trees, graphs) related through proximity data, as a low-dimensional embedding can help in visualizing the abstract space. Embedding can also be applied when the objects are points in a vector space whose dimensionality is too large for the application of data analysis algorithms, such as clustering. In such cases, embedding can be used to lower the dimensionality of the space.

1.1 Background

In general, embedding techniques fall into two categories: linear and nonlinear. Classical **linear embedding**, as embodied by principal component analysis (PCA), reduces dimensionality by projecting high-dimensional data onto a low-dimensional subspace. The optimal *p*-dimensional subspace is selected by rotating the coordinate axes to coincide with the eigenvectors of the sample covariance matrix, and keeping the *p* axes along which the sample has the largest variance. Principal component analysis directly applies to data that already resides in a real normed space. It can also be applied to proximity data that has been appropriately preprocessed, under certain spectral conditions on the matrix of pairwise distances (Cox and Cox, 2001).

Nonlinear embedding techniques, also referred to as multidimensional scaling (MDS) techniques, apply to a broad set of data types. Generally speaking, the goal of MDS is to construct a low-dimensional map in which the distance between any two objects corresponds to their degree of dissimilarity. The method maps a given set of samples into a space of desired dimension and norm. A random mapping (or projection by PCA) can serve as the initial embedding. A stress function that compares proximity values with distances between points in the host space (usually a sum-of-squared-errors function) is used to measure the quality of the embedding, and a gradient descent procedure is applied to improve the embedding until a local minimum of the stress function is reached. Like PCA, MDS attempts to preserve all pairwise distances as well as possible; but the restriction to linear projections is removed, and arbitrary embeddings are considered. Many variants of this general approach are reported in the literature; a broad overview of the field is given by Cox and Cox (2001).

The MDS method was traditionally used to visualize high-dimensional data in two or three dimensions. It has long been employed for data analysis in the social sciences, where the generated maps tend to have only a few hundred data points, and computational efficiency is not a factor (Sammon, 1969). Practically, such procedures are not effective for more than few thousand sample points. More recently, MDS has been turned toward the visualization of large biological and chemical data sets, with thousands or even millions of points (Yona, 1999; Apostol and Szpankowski, 1999). Applying traditional MDS to very large data sets is prohibitively slow, leading several authors to propose approximations and workarounds. Linial et al. (1995) presented a randomized approach that attempts to bound the distortion. However, the bound is not tight, and in practice this approach can introduce large distortions, as no objective function is explicitly optimized. A different randomized approach, based on iteratively adjusting the lengths of randomly selected edges, was proposed by Agrafiotis and Xu (2002). This method has linear time complexity, and is therefore well-suited to extremely large data sets. Basalaj (1999) proposed an incremental method for large-scale MDS. It consists of embedding a small subset of objects carefully, then using this skeleton embedding to determine the positions of the remaining objects.

Recently, a new class of non-linear embedding techniques has emerged: the **manifold learning** algorithms, which comprise an active area of research. These algorithms are designed to discover the structure of high-dimensional data that lies on or near a low-dimensional manifold. There are several approaches. The Isomap algorithm (Tenenbaum et al., 2000) uses geodesic distances between points instead of simply taking Euclidean distances, thus “encoding” the manifold structure of the input space into the distances. The geodesic distances are computed by constructing a sparse graph in which each node is connected only to its closest neighbors. The geodesic distance between each pair of nodes is taken to be the length of the shortest path in the graph that connects them. These approximate geodesic distances are then used as input to classical MDS. The LLE algorithm (Roweis and Saul, 2000; Saul and Roweis, 2003) uses a collection of local neighborhoods to guide the embedding. The assumption is that if the neighborhoods are small, they can be approximated as linear manifolds, and the position of each point can be reconstructed as a weighted linear combination of its k nearest neighbors. The positions of the points in the lower-dimensional space are determined by minimizing the reconstruction error in this low-dimensional space (with fixed weights that were determined in the original high-dimensional space). This is done by solving an eigenvector problem, as in PCA. Another approach is the eigenmaps method. The goal of this type of method is to minimize a quadratic form (either the squared Hessian or the squared gradient) over all functions mapping the manifold into the embedding space (Donoho and Grimes, 2003; Belkin and Niyogi, 2002). When the continuous function is approximated by a linear operator on the neighbor graph, the maximization problem becomes a sparse matrix eigenvalue problem and is readily solved.

The manifold learning methods form a powerful generalization of PCA. Unlike PCA, which is useful only when the data lies near a low-dimensional *plane*, these methods are effective for a large variety of manifolds. By using a collection of local neighborhoods, or by exploiting the spectral properties of the adjacency graph, they extract information about local manifolds from which the global geometry of the manifold can be reconstructed. In practice, preserving these local manifolds results in non-linear embeddings. The underlying principles of these methods are similar, and their power stems from the fact that they practically employ alternative representations for the data points. PCA seeks correlation between features and represents the data best in a sum-of-squared-errors sense. However, it implicitly assumes the Euclidean metric. On the other hand, the manifold learning algorithms explore the properties of the adjacency graph to form a new representation, inducing a new metric. For example, the geodesic distance in essence samples the geometry of the input manifold, and it is that definition to which one can attribute the great success of the Isomap algorithm. Similarly, the spectral approaches use the proximity data to derive the new representation that reflects collective properties. This is related to other studies that showed that encoding data through collective or transitive relations can be very effective for data representation (e.g., embedding) as well as for clustering (Smith, 1993; Wu and Leahy, 1993; Shi and Malik, 1997; Blatt et al., 1997; Gdalyahu et al., 1999; Dubnov et al., 2002).

The different types of embedding methods are inherently suited to different types of problems. PCA identifies significant coordinates and linear correlations in the original, high-dimensional data. It is therefore appropriate for finding a simple, linear, globally applicable rule for extracting information from *new* data points. It is unsuitable when the correlations are nonlinear or when no simple rule exists. General multidimensional scaling techniques are appropriate when the data is highly nonmetric and/or sparse. However, MDS is iterative, does not guarantee optimality or uniqueness of its output, does not generate a rule for interpreting new data, and is typically quite slow compared with other methods. These deficiencies are only tolerable when weighed against the greater generality and simpler formulation of multidimensional scaling. Finally, manifold-learning techniques are appropriate when a strong nonlinear relation exists in the original data. In such cases, the methods described can make use of powerful, noniterative methods, with guaranteed global optimality. They are less suitable when not enough data is available, or when the data points are inconsistent with a manifold topology (for instance, lying on a structure with branches and loops), or when the data is intrinsically nonmetric.

1.2 Method

The algorithm presented in this paper is in the class of nonlinear embedding techniques. However, unlike the manifold learning methods, our focus is on the higher-order structure of the data. The aforementioned approaches optimize an objective function that is a function of the individual pairwise distances or their derivatives. However, collective aspects of the embedding are not explicitly considered, even when local neighborhoods are used. This problem is addressed in this paper.

In a recent study by Roth et al. (2002), the authors point out that high-dimensional PCA, applied to dissimilarity data that has been shifted by an additive constant, automatically preserves some clustering properties of the original data. Specifically, they show that the optimal partition of the original data points into *k* clusters (using a particular cost function, which they define) is identical to the optimal partition of the embedded data points, using the standard *k*-means cost function. However, a subsequent reduction in the embedding dimension is often desirable, and the clustering properties are not preserved (or even considered) in this second stage.

Our interest in embedding algorithms emerges from our even stronger interest in studying high-order organization in complex spaces. In a typical application one is interested in exploratory data analysis, discovering patterns and “functional” meaningful clusters in the data. Embedding is often used to visualize complex data in a low-dimensional space, in the hope that it will be easier to discover structure or statistical regularities in the reduced data. Thus, optimal embedding should consider not only the distortion in pairwise distances that is introduced by the embedding, but also the **geometric distortion**, i.e., the disagreement on the intrinsic structure of the data. Finding the optimal embedding thus becomes a problem of optimizing a complex criterion function that seeks to jointly improve both aspects of an embedding. Our approach tackles the problem from this perspective and attempts to preserve these patterns by implicitly encoding the cluster structure into the cost function. Here we present for the first time such a criterion function and describe the means to optimize it.

Another new element of our paper is a method to deduce the right dimension for the data. Existing methods for dimensionality reduction are looking for elbows in the residual variance graph to determine the right dimensionality, however, the exact definition is subjective and qualitative. Here we introduce two quantitative methods to deduce the right dimension.

The paper is organized as follows. We first describe the two commonly-used MDS objective functions, the SAMMON and SSTRESS functions, and present improved algorithms for optimizing them. Next, we present a hierarchical method for efficiently embedding data sets that consist of many subsets or clusters of related objects. We then present the main element of this paper, a new type of MDS called **distributional scaling**, which directly addresses the problem of structure preservation during the embedding process. Distributional scaling strives to maintain the *distribution* of dissimilarities, as well as the individual dissimilarities themselves, thereby using higher-order information to create a more informative map. Next, we describe two distinct methods for ascertaining the best dimension in which to embed a given data set. Finally, we test the performance of distributional MDS on a large number of synthetic data sets. By using this new form of scaling, we demonstrate that we are able to remove undesirable artifacts from embeddings produced by traditional MDS.

2. Theory

We start with some basic definitions and a review of classical metric and nonmetric MDS. We then introduce hierarchical MDS and Distributional MDS, and discuss the measures that we use to evaluate similarity between probability distributions. We conclude this section with a method to choose the embedding dimension.

2.1 Definition and Mathematical Preliminaries

Throughout this paper we will be interested in optimizing embeddings of sets of objects in Euclidean space. An embedding of n objects in p-dimensional Euclidean space is a set of image points $\mathbf{x}_i \in \mathcal{R}^p$, where $i = 1, \ldots, n$. We take S_n^p to be the set of all such embeddings.

We primarily will be interested not in the image points themselves, but in the distances between them. Let Ω_n be the set of symmetric $n \times n$ matrices with zeros along the diagonal. For each embedding X of n objects, we can define the distance matrix $D(X) \in \Omega_n$, with matrix elements $D_{ij} = ||\mathbf{x}_i - \mathbf{x}_j||$. Since the interpoint distances are invariant under Euclidean transformations of the entire configuration of points (that is, translations, rotations, and reflections), D is many-to-one. We denote by D_n^p the image of S_n^p under the mapping D. This is the space of all possible distance matrices arising from p-dimensional embeddings of n points.

Formally, the optimization problem is defined as follow: we are given a set of n objects and their dissimilarities. Denote by Δ_{ij} the dissimilarity of objects i and j. The goal is to find a configuration of image points $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n$ such that the $n(n-1)/2$ distances D_{ij} between image points are as close as possible to the corresponding original dissimilarities Δ_{ij}.

2.2 Metric MDS

The simplest case is **metric MDS**, where the dissimilarity data is quantitative. We are given n objects, together with a target dissimilarity matrix $\Delta \in \Omega_n$. The goal is to find an embedding X such that the distance matrix $D(X)$ matches Δ as closely as possible. This is formulated as a weighted least-squares optimization problem: given $(\Delta, W) \in \Omega_n \times \Omega_n$, where $W = (w_{ij})$ is a symmetric matrix of weights, minimize

$$\mathcal{H}(X) = \sum_{i<j} w_{ij} \Big(f(D_{ij}(X)) - g(\Delta_{ij}) \Big)^2 \tag{1}$$

over all $X \in S_n^p$. The functions f and g determine exactly how errors are penalized. Two common choices for these functions are considered here. The stress, or SAMMON, objective function is defined by $f(x) = g(x) = x$. The squared stress, or SSTRESS, function is defined by $f(x) = g(x) = x^2$.

$$\begin{aligned}
\mathsf{SAMMON}: \quad \mathcal{H}(X) &= \sum_{i<j} w_{ij} \Big(D_{ij} - \Delta_{ij} \Big)^2 , \\
\mathsf{SSTRESS}: \quad \mathcal{H}(X) &= \sum_{i<j} w_{ij} \Big(D_{ij}^2 - \Delta_{ij}^2 \Big)^2 .
\end{aligned}$$

The SAMMON and SSTRESS objective functions have somewhat different advantages. While the former seems more natural, being the square of the Euclidean metric in Ω_n, and may produce more aesthetically pleasing embeddings, the latter is more tractable from a computational standpoint, and seemingly less plagued by nonglobal minima (Malone and Trosset, 2000).

The weights contained in the weight matrix W are arbitrary. They can be used to exclude missing proximity data, or to account for data with varying confidence levels. In practice, however, the weights are often defined in terms of Δ. Three choices of this type are:

$$\begin{aligned} w_{ij}^{-1} &= \sum_{m<n} g(\Delta_{mn})^2 \,, \\ w_{ij}^{-1} &= g(\Delta_{ij}) \sum_{m<n} g(\Delta_{mn}) \,, \\ w_{ij}^{-1} &= \frac{1}{2} n(n-1) g(\Delta_{ij})^2 \,. \end{aligned}$$

All three choices normalize the metric stress function, in the sense that $\mathcal{H}(0) = 1$. We refer to the first one as global weighting, the second as intermediate (or semilocal) weighting, and the third as local weighting. Unless otherwise specified, the global weighting scheme is used in this paper.

The numerical optimization of the metric stress function is not entirely trivial. The deterministic algorithms (gradient descent) that are typically applied to solve this problem converge to local minima, which may not be globally optimal. It is possible to use stochastic techniques, like simulated annealing (Klein and Dubes, 1989), to reduce or eliminate the probability of being trapped in a nonglobal minimum, albeit at the cost of increased computation time. Recently, Klock and Buhmann (1997) have demonstrated that so-called *deterministic annealing* can be used to avoid poor minima without sacrificing too much efficiency, thus combining the merits of the stochastic and deterministic approaches. Such globalization strategies are outside the scope of the present study. Instead, we have developed an efficient method for finding *local* minima that takes advantage of special features of the SSTRESS and SAMMON objective functions. This algorithm is described in detail in Appendix A.

2.3 Nonmetric MDS

A generalization of the metric problem is **nonmetric MDS**, which is appropriate when the dissimilarity data is not quantitative, but merely ordered. In this case, we minimize an objective function like Eq. (1) over X, while also allowing g to vary over all increasing functions. As with metric MDS, the Euclidean distances will be transformed by a known function $f(x)$, which we will restrict to be x or x^2, in the SAMMON and SSTRESS cases respectively.

Note that, if we were to use Eq. (1) with fixed weights, the objective function would be trivially minimized by taking g to zero and shrinking the configuration X to a single point. Instead we use global weighting, as described above. This sets the overall weight to an appropriate functional of g, producing a scale-invariant objective function:

$$\mathcal{H}_{nm}(X,g) = \frac{\sum_{i<j} \Big(f(D_{ij}(X)) - g(\Delta_{ij}) \Big)^2}{\sum_{i<j} g(\Delta_{ij})^2} \,. \tag{2}$$

Our algorithm for optimizing metric MDS can be extended to cover the nonmetric case as well. Appendix B discusses the necessary modifications.

2.4 Hierarchical MDS

In many cases the data is naturally organized in classes that have subclasses, that are composed of subsubclasses, and so on. Such a hierarchical classification can be obtained either externally or by applying data analysis techniques, such as clustering.

When the points to be embedded are pre-grouped into clusters, it is natural to treat the measured dissimilarities *between* clusters differently from those *within* a particular cluster. The task of finding a good global embedding splits into two subtasks: (a) finding a good embedding for each individual cluster, and (b) ensuring that these embedded clusters are well-placed with respect to each other. For a cluster that can be further divided into subclusters, step (a) can be performed recursively. For clusters that cannot be further divided, step (a) is carried out with ordinary metric or nonmetric MDS, or with the distributional scaling technique we will introduce in a subsequent section. We refer to this procedure as **hierarchical MDS**.

It remains to specify the details of step (b), the placement of embedded clusters with respect to each other. This is done by searching for a transformation that will minimize the overall stress, now considering all intercluster distances, as well as the intracluster distances that are already optimized. Clearly, clusters should be allowed to undergo arbitrary Euclidean transformations, as these do not increase their internal stress. The Euclidean transformations of $\mathcal{R}^p$ are parametrized by a p-vector $\mathbf{X}$ and an orthogonal $p \times p$ matrix M, and act on an arbitrary point $\mathbf{y}$ as $\mathcal{E}_{\mathbf{X},M}(\mathbf{y}) = M \cdot \mathbf{y} + \mathbf{X}$. We choose to allow, more generally, all affine transformations. The affine transformations are parametrized in the same way, except that M need not be orthogonal. The space of affine transformations is a linear subspace of the full search space, thus simplifying the search. Moreover, the space of affine transformations is *connected*, unlike the space of Euclidean transformations.

Formally, we are given a partitioning of the target points into K clusters, and an initial embedding that was carried out for each cluster individually. Let $\{\mathbf{y}_i\}$ be the initial coordinates of the points in cluster A. We stipulate that the final coordinates $\{\mathbf{x}_i\}$ are generated by *affine* transformations of these single-cluster embeddings, where each cluster is transformed independently. That is, the final coordinates of point $i \in A$ are given by

$$\mathbf{x}_i = \mathbf{X}_A + M_A \cdot \mathbf{y}_i$$

for some affine transformation $(\mathbf{X}_A, M_A)$. Our final embedding is generated by minimizing the overall metric stress, allowing only the $(\mathbf{X}, M)$ pairs to vary, while the base coordinates $\mathbf{y}_i$ are held fixed. That is, individual clusters can be rotated and translated with respect to each other, and stretched in a small number of ways; but they cannot be split into two or otherwise fundamentally reshaped.

Restricting the allowed configurations in this way reduces the number of degrees of freedom enormously. For instance, an arbitrary two-dimensional embedding of 100 points requires 200 parameters for its description, while an arbitrary affine transformation of a *known* two-dimensional embedding requires only 6. This reduction helps us in two ways. First, optimization within a subspace usually converges much faster simply because the search space is smaller. Second, we may be able to streamline the evaluation of the objective function $\mathcal{H}$ once we have fixed the coordinates $\mathbf{y}_i$. For SSTRESS, this can be done exactly, by rewriting the stress function in terms of the $\mathbf{X}_A$ and M_A variables. Specifically, when the final coordinates $\mathbf{x}_i$ are restricted to affine images of a known base embedding $\mathbf{y}_i$, the SSTRESS function becomes

$$\mathcal{H} = \sum_{i,j} w_{ij} \left(||\mathbf{x}_i - \mathbf{x}_j||^2 - \Delta_{ij}^2 \right)^2$$

$$\begin{aligned}
&= \sum_{A,B} \sum_{i \in A, j \in B} w_{ij} \left(||\mathbf{X}_A - \mathbf{X}_B + M_A \cdot \mathbf{y}_i - M_B \cdot \mathbf{y}_j||^2 - \Delta_{ij}^2 \right)^2 \\
&= \sum_{A,B} \sum_{\alpha,\beta} \sum_{i \in A, j \in B} w_{ij} \left((M_A^T M_A)_{\alpha\beta} (\mathbf{y}_i)_\alpha (\mathbf{y}_i)_\beta + \ldots + ||\mathbf{X}_A - \mathbf{X}_B||^2 - \Delta_{ij}^2 \right)^2 \\
&= \sum_{A,B} \sum_{\alpha,\beta,\gamma,\delta} P^{(AB)}_{\alpha\beta\gamma\delta} (M_A^T M_A)_{\alpha\beta} (M_A^T M_A)_{\gamma\delta} + \ldots + W^{(AB)} , \qquad (3)
\end{aligned}$$

where many terms are omitted for brevity. Partial sums over ij have been performed wherever possible, leading to parameters that can be computed in advance, such as

$$\begin{aligned}
P^{(AB)}_{\alpha\beta\gamma\delta} &= \sum_{i \in A, j \in B} w_{ij} (\mathbf{y}_i)_\alpha (\mathbf{y}_i)_\beta (\mathbf{y}_i)_\gamma (\mathbf{y}_i)_\delta , \\
W^{(AB)} &= \sum_{i \in A, j \in B} w_{ij} \Delta_{ij}^4 ,
\end{aligned}$$

and so on. The rewritten SSTRESS function is a complicated expression, but it contains a relatively small number of terms. Specifically, for an embedding problem in p dimensions, involving K clusters with N points each, the new expression is a sum over $O(K^2 p^4)$ terms, while the original metric stress function has $O(K^2 N^2)$ terms. The upshot is that for large clusters, with $N \gg p^2$ points apiece, using Eq. (3) can save computational labor.

Most importantly, hierarchical MDS proved most effective for highly frustrated data, or when embedding high dimensional data in low dimension. In such cases direct embedding of the complete data set tends to diminish any high-order structure that exists in the data, while hierarchical MDS preserves more of the structure.

2.5 Introducing Distributional MDS

Metric MDS, as defined in the previous sections, works well in many cases. When the metric stress of an embedding is sufficiently low, one knows that all embedded edges are close to their target lengths, and hence the input data is well-represented by the final map. However, cases arise in which *no* embedding has an acceptably low level of stress.[1] In such cases, the precise *quantitative* structure of the input data is impossible to maintain, and the metric stress alone does not distinguish between qualitatively good and bad maps.

An illustrative example, which will serve as our motivation for introducing a new type of multidimensional scaling, is shown in Figure 1. It depicts an embedding of 600 points in two dimensions, generated by applying metric SSTRESS to synthetic, random proximity data.[2] The points were originally sampled from three clusters, such that the distances between clusters tend to be greater than those within clusters, as described in the figure caption. However, as seen in the figure, the process of embedding splits the central cluster into two well-separated subclusters. This is purely an artifact of the metric scaling process, as there is no inherent difference between the points in the two subclusters. Moreover, the partitioning into subclusters is not robust, but differs from run to run

[1] The amount of acceptable stress will vary from application to application and also depends on the demands of the user.

[2] Note that this data is nonmetric, since the triangle inequality does not hold, and that it is represented only by its proximity matrix. This kind of data arises naturally in cases where the objects are abstract or difficult to map to a vector space (e.g., strings, graphs, biological macromolecules, DNA and protein sequences).

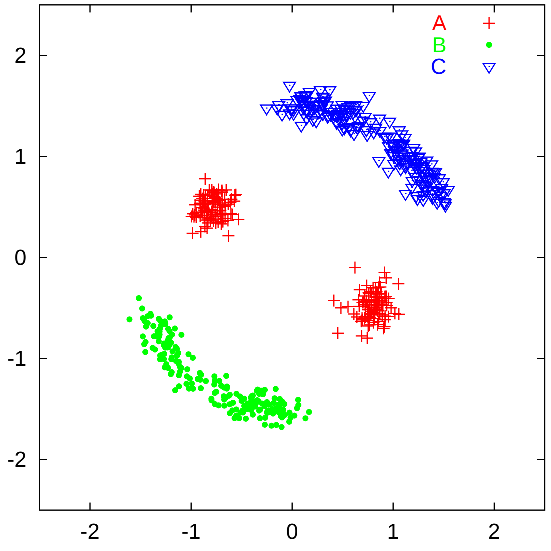

Figure 1: **Structural artifact generated by metric SSTRESS.** Three 200-point clusters (*A*, *B*, and *C*) were embedded in two dimensions using metric MDS and a synthetic dissimilarity matrix Δ. The central cluster (*A*) has been split into two apparent subclusters by the embedding process. To generate Δ, each target dissimilarity Δ_{ij} was drawn from one of three chi distributions. If i and j are in the same cluster, $\Delta_{ij} \sim \chi_2(1.0)$. If ij connects cluster *A* to cluster *B* or *C*, then $\Delta_{ij} \sim \chi_2(1.5)$. Finally, if ij connects clusters *B* and *C*, then $\Delta_{ij} \sim \chi_2(2.0)$.

when random starting configurations are used. Similar results can be obtained with the SAMMON criterion function, and with nonmetric MDS. This is a dramatic type of artifact, which we would like to automatically diagnose and avoid.

Our goal is to produce embeddings that preserve some notion of structure over the input space. The concept of geometry might not be clearly defined for the input space, and since the data set may be non-Euclidean or even nonmetric, it is hard to speak in general terms about the structure of the data. In our study we focus on the clustering properties of the data. The cluster structure reflects the existence of inherent order and the presence of groups and subgroups that usually can be mapped to specific subcategories of the data (for example, functional, topological, or demographic, depending on the data set). It is this notion of order that we would like to preserve. Thus, in our case, the definition of similar structures relies on the clustering profile of the data.

One way to characterize the underlying cluster structure of data is by studying the distribution of distances between and within clusters.[3] Although similar distributions do not guarantee that the embedding will have the same clustering profile, it reduces the search space to embeddings that are more likely to have the same structure. The simple example of Figure 1 demonstrates this point. Figure 2 shows histograms of the set of interpoint distances, both before and after the embedding process. From these graphs it is clear that the embedding has qualitatively altered the information present: although the target distances form a unimodal distribution, the post-embedding curve is distinctly bimodal. There is evidence that this kind of artifact is also prevalent in real applications of metric MDS (Yona, 1999).

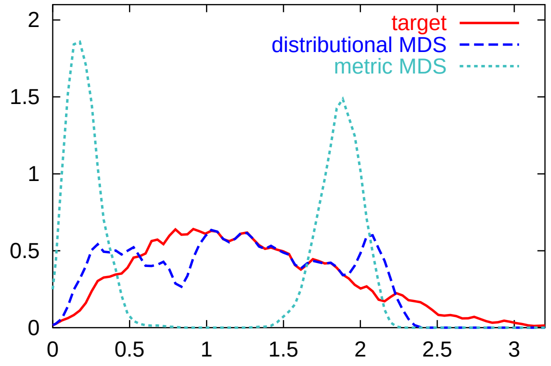

Figure 2: **Distribution of interpoint distances within a split cluster.** The three curves represent the distributions of the target distances Δ_{ij} (see the caption to Figure 1), the embedded distances D_{ij} from metric SSTRESS, and the embedded distances D_{ij} from our proposed distributional scaling. The two-dimensional embeddings from metric and distributional MDS are shown in Figure 1 and Figure 3, respectively.

To correct for artifacts of this type, and more generally to preserve the structural information we have just discussed, we propose a modified objective function that penalizes discrepancies like that shown in Figure 2. This new objective function can be used whenever cluster assignments are known, or can be estimated. For each pair of clusters, A and B, we define ρ_{AB} to be the (weighted, normalized) distribution of embedded distances between the points in cluster A and those in cluster B:

$$\rho_{AB}(x) = \frac{\sum_{i\in A}\sum_{j\in B} w_{ij}\delta(x - D_{ij})}{\sum_{i\in A}\sum_{j\in B} w_{ij}} \,. \tag{4}$$

Here $\delta(x)$ is the Dirac delta function, which describes a point mass of weight 1 localized at the origin. Similarly, we denote by $\tilde{\rho}_{AB}$ the distribution of the A–B target distances (the elements of Δ). Our proposed new objective function has the general form

$$\mathcal{H}_d(X) = (1-\alpha)\mathcal{H}(X) + \alpha \sum_{A\leq B} W_{AB} D[\rho_{AB}, \tilde{\rho}_{AB}] \,, \tag{5}$$

[3] Preserving just the cluster assignments, as is done by Roth et al. (2002), might miss higher order structure over clusters. Moreover, the method proposed by Roth is algorithm-dependent (tailored to the k-means algorithm).

where $D[p,q]$ is some measure of the dissimilarity between two distributions, the W_{AB} are relative weights of the target distributions, and α determines the balance between the original metric component of the stress and this new, distribution-related, component. We call the optimization of this type of objective function **distributional MDS**.

One could use any number of other measures to represent the data structure and its geometry. For example, cluster diameters, or the first and second moments of the sample points in each cluster, could be used in addition to the distributions of pairwise distances. The objective function could be modified to include these (or other, data-specific) order parameters. Rather than attempting to include all possible choices, we chose the single-parameter form of $\mathcal{H}_d$ given above. For the dissimilarity measure D, we will use the earth-mover's distance, a metric which is motivated and described in Section 2.6.

The weights W_{AB} are assigned based on the information content of the distributions. Specifically, we use the entropy

$$S_{AB} = S[\tilde{\rho}_{AB}] \equiv -\int dx\, \tilde{\rho}_{AB}(x) \log_2 \tilde{\rho}_{AB}(x)$$

as a measure for the information content of the target distribution, and we set $W_{AB} = 2^{-S_{AB}}$. Thus, the lower the entropy of a distribution, the more significant the contribution of that term to the objective function. Our motivation for this choice of weights is heuristic: high-entropy distributions are more likely to arise by chance, while low-entropy distributions are more likely to reflect a true pattern in the data. With robustness to classification errors in mind (see below), this weighting scheme attempts to minimize the sensitivity of the model to noise by emphasizing the low-entropy target distributions.

It is important to note that the availability of cluster information is by no means a hurdle or a limiting factor of this algorithm. One can use any sensible clustering algorithm (e.g., k-means), applied to the original data or to its metric embedding, to suggest a preliminary classification. If the data is sufficiently ordered, this clustering profile can provide a rough snapshot of the geometry, the quality of which depends on the clustering algorithm and the data set.[4] This clustering profile can then be used to guide the embedding process, even if it is not completely accurate. Since the distributions between all pairs of clusters are considered, the algorithm avoids embeddings that grossly distort the cluster structure, even when the higher-order structure of the data is misrepresented (e.g., when a real cluster is split into two by a clustering algorithm).

To demonstrate, we return to the previous example, supposing now that the true cluster assignments are unknown. Applying k-means clustering (with both $k = 4$ and $k = 3$) to the metric embedding (Figure 1) produces the tentative classifications shown in Figure 4. The k-means results exhibit both **overclassification**, where a single true cluster is broken into two classes, and **misclassification**, where parts of two true clusters are combined in a single class. Applying distributional scaling to the original dissimilarities, using the *tentative* cluster assignments rather than the true ones, produces the improved embeddings shown in Figure 5. Both resemble the embedding in Figure 3, which was generated using the true assignments. This is a satisfying result, since it indicates that our algorithm is robust with respect to at least some classification errors. In general, the prob-

[4] Given the choice between a conservative clustering and a more permissive one (e.g., hierarchical clustering with different thresholds), one might prefer the conservative algorithm. Note that here we ignore issues of generalization and model validity of the clustering profile, as they are irrelevant at this point. Opting for structure-preserving embeddings, smaller and more compact clusters can be considered as entities of high confidence and are more amenable to undergo this process successfully.

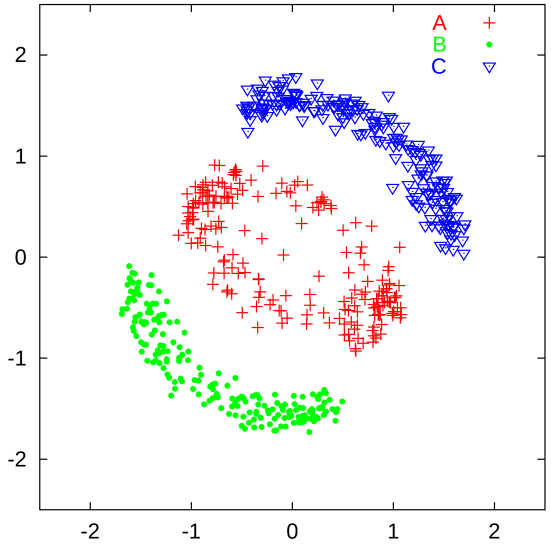

Figure 3: **Improved map from distributional scaling.** Starting from the metric embedding, the objective function defined by Eq. (5) (with $\alpha = 0.1$) was numerically optimized. The artifact seen in Figure 1 is largely corrected: cluster *A* now appears as a single cluster, as it should.

lem of overclassification is well-corrected by our algorithm. The problem of misclassification is not addressed as well; but in cases like the example, where intercluster and intracluster distances have substantially different distributions, distributional MDS gives a more reliable picture of the actual data than metric MDS alone.

2.6 The Earth-Mover's Distance Between Probability Distributions

There are several common measures to assess the statistical similarity of probability distributions, among which are the Manhattan distance (the L_1 norm) and the KL divergence (Kullback, 1959). Our first choice was the information-theoretic Jensen-Shannon divergence measure (Lin, 1991), which is a symmetric and bounded variant of the KL divergence. Formally, given two (empirical) probability distributions $\mathbf{p}$ and $\mathbf{q}$, for every $0 \leq \lambda \leq 1$, the **λ-JS divergence** is defined as

$$D_{\lambda}^{JS}[\mathbf{p}||\mathbf{q}] = \lambda D^{KL}[\mathbf{p}||\mathbf{r}] + (1-\lambda) D^{KL}[\mathbf{q}||\mathbf{r}] \ ,$$

where $D^{KL}[\mathbf{p}||\mathbf{q}] = \sum_i p_i \log_2(p_i/q_i)$ is the KL divergence, and $\mathbf{r} = \lambda\mathbf{p} + (1-\lambda)\mathbf{q}$ can be considered as the most likely common source distribution of both distributions $\mathbf{p}$ and $\mathbf{q}$, with λ as a prior weight. The parameter λ reflects the *a priori* information and is set by default to 0.5.

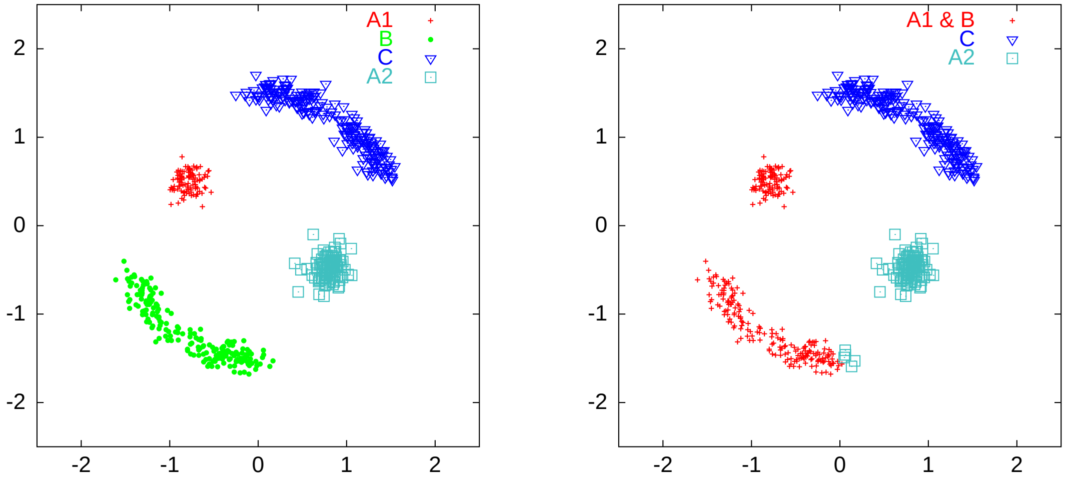

Figure 4: **Naive cluster assignments** generated by the application of k-means clustering ($k = 4$, left; $k = 3$, right) to the points in Figure 1. Note that the true cluster assignments were never used. The $k = 4$ example shows **overclassification**, where a single true cluster is broken into two classes. The $k = 3$ example shows **misclassification**, where parts of two true clusters are combined in a single class.

Despite its attractive properties as a measure of statistical similarity,[5] we learned quite early on that this measure is inappropriate when attempting to preserve the overall shape of the distribution. Specifically, this measure was found to be difficult to optimize through a local search. Since the Jensen-Shannon distance is a purely local measure of the difference between two distributions, a JS-based algorithm is easily trapped in poor local minima.

A more effective measure of dissimilarity between two distributions is the earth-mover's distance (EMD) (Rubner et al., 1998). As shown in Figure 6, the EMD is substantially easier to minimize than the Jensen-Shannon divergence. Given two probability distributions p and q over the interval $[0,K]$ (which can be thought of as distributions of "earth" and "holes" respectively), the EMD between p and q can be defined by means of the following transport or bipartite-graph flow problem. Let $f(x,y)$ be the amount of earth (flow) carried from $x \in [0,K]$ to $y \in [0,K]$, such that every hole is filled and no new holes are dug. In other words, $f(x,y)$ is a flow function that should satisfy

$$\begin{aligned} f(x,y) &\geq 0\,, \\ p(x) &= \int_0^K dy\, f(x,y)\,, \end{aligned}$$

[5] Besides being bounded and symmetric, it has been shown that the JS divergence measure is proportional to minus the logarithm of the probability that the two empirical distributions represent samples drawn from the same ("common") source distribution (El-Yaniv et al., 1998).

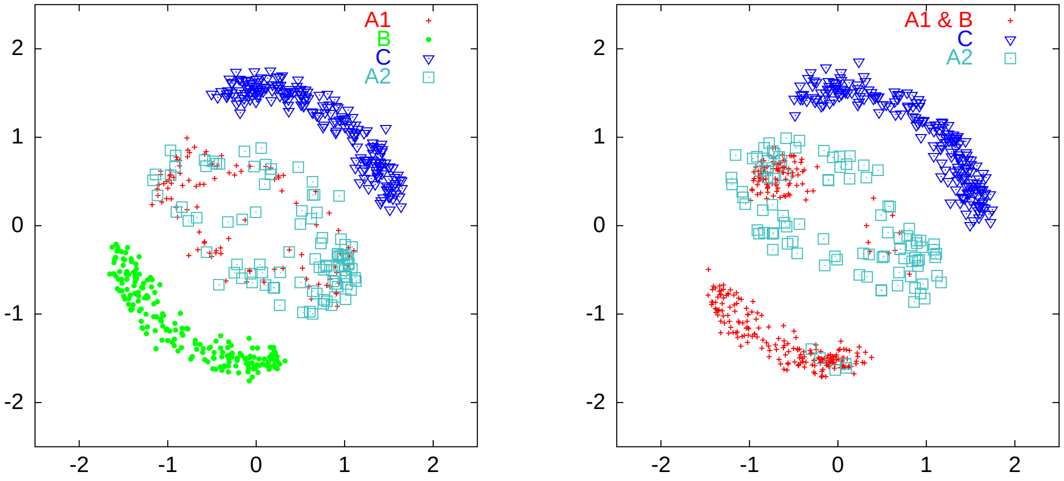

Figure 5: **Distributional scaling with naive cluster assignments.** The figure was generated in the same way as Figure 3, except that the *k*-means cluster assignments (from Figure 4) were used in place of the true ones. In both examples, the process merges the two central groups of points, while keeping them separate from the remaining two groups.

$$q(y) \quad = \quad \int_0^K dx\, f(x,y)\,.$$

Let $dist(x,y)$ be the "ground distance" between x and y. (In our case, $dist(x,y) = |x-y|$.) Then the EMD is the *minimum* total distance traveled by the earth,

$$\text{EMD}\,[p,q] = \min_f \int dx \int dy\, dist(x,y)\, f(x,y)\,,$$

subject to the given constraints on f. Intuitively, the EMD can be considered as the minimal amount of work required to match p with q. It can be shown that the EMD between normalized one-dimensional distributions is the same as the L_1 distance between their *cumulative* distribution functions (Levina and Bickel, 2001). That is, the earth-mover's distance between distributions p and q is just

$$\text{EMD}\,[p,q] = \int_0^K dx \left| \int_0^x dy (p(y) - q(y)) \right| \,.$$

The result follows from the fact that there is a greedy algorithm for finding the minimal flow in one dimension (only): fill the leftmost unfilled hole with the leftmost available dirt until all holes are filled. This expression is essential for our algorithm, since it makes the EMD simple to calculate and differentiate, rendering it suitable for inclusion in the stress function.

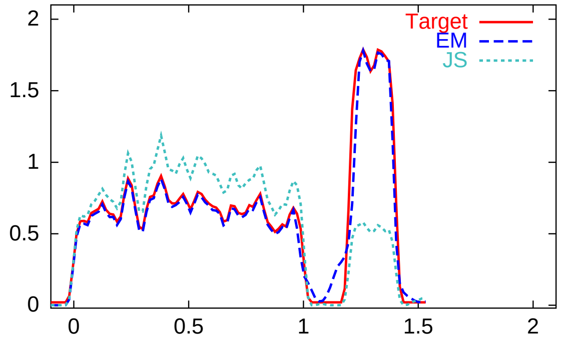

Figure 6: **Comparison of EMD-based algorithm with Jensen-Shannon algorithm.** The elements of a 200 × 200 dissimilarity matrix were drawn from a bimodal distribution ("Target"). Downhill search was used to find a two-dimensional embedding with interpoint distance distribution closest to the target distribution, under the EMD and JS measures. The JS-based algorithm became trapped in a local minimum: shifting weight to the right does not immediately decrease the Jensen-Shannon distance between the target and JS curves. The EMD-based algorithm, on the other hand, reproduced the target distribution accurately.

2.6.1 IMPLEMENTATION ISSUES

The naive implementation of the algorithm is impractical for large data sets, because building and storing the exact, discrete distribution defined by Eq. (4) takes a large amount of space, $O(n^2)$, and calculating the EMD between two such distributions takes $O(n^2 \log n)$ time.[6] Moreover, the earth-mover's distance between two such distributions has many nondifferentiable points along any given line, which is problematic for our (gradient-based) optimization strategy. We address both these issues by using an approximate distribution in place of the exact one.

Our approximate distributions are piecewise constant, consisting of a relatively small number of disjoint bins. We associate the k-th bin with the interval $[x_k, x_{k+1}]$. To build the necessary distribution, each delta-function in Eq. (4) is first broadened into a finite-width shape with the correct total weight. That is, $a\delta(x-b) \rightarrow ah(x-b)$, where h is a smooth function. The weight is then distributed among the relevant bins: the k-th bin is incremented by $a\int_{x_k}^{x_{k+1}} h(x-b)dx$ (see Figure 7). The result is a histogram whose bin contents are differentiable functions of the distances D_{ij}. We use this histogram in our calculation of the earth-mover's distance; using the chain rule, the EMD is then differentiable as well.[7]

[6] The rate-limiting step is the sorting of the values D_{ij}, which is needed to find the cumulative distribution function.

[7] More precisely, the EMD still has nondifferentiable points, but they are sparse enough that there are none on a typical ray in the search space.

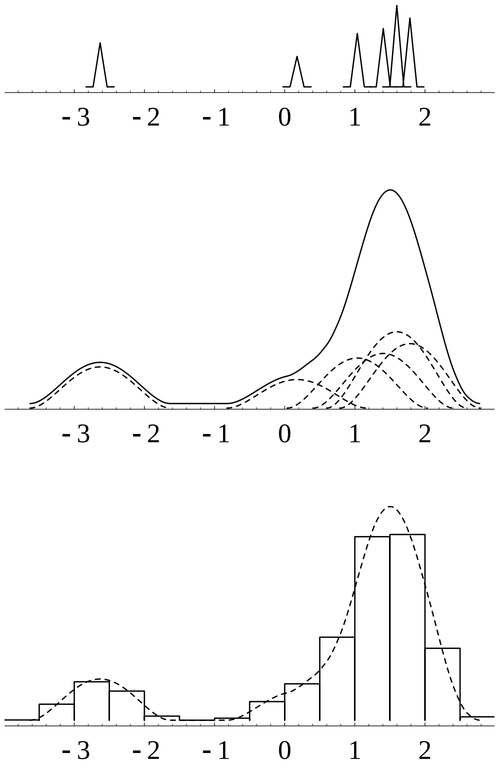

Figure 7: **Histogram construction.** To construct a histogram from a discrete distribution (top), we first broaden each point by a smooth window function (middle). The integrated weights are then used as the bin counts for the histogram (bottom).

2.7 Choosing the Initial Embedding

Since our optimization method is iterative, beginning with a low-stress embedding can save time and, potentially, improve the final result. We suggest two inexpensive ways to generate a reasonably good initial configuration.

The first method is principal component analysis (PCA). Principal component analysis is well known as the basis for classical scaling (Young and Householder, 1938; Gower, 1966); given a data set with a low-stress embedding, PCA can be used to find a good configuration very quickly. To find the principal components we first form the auxiliary matrix M, with matrix elements

$$M_{ij} = -\frac{1}{2}\Delta_{ij}^2 + \frac{1}{2n}\sum_k \left(\Delta_{ik}^2 + \Delta_{jk}^2\right) - \frac{1}{2n^2}\sum_{k,l}\Delta_{kl}^2 \; . \tag{6}$$

To generate an embedding in p dimensions we compute the p largest eigenvalues of M, together with their associated eigenvectors (λ_a and $\mathbf{u}_a$). Finally, we form an initial configuration with coordinate components

$$(\mathbf{x}_i)_a = \sqrt{\lambda_a}(\mathbf{u}_a)_i$$

for $a = 1, \ldots p$. If Δ is, in fact, a distance matrix $D(X)$ with $X \in S_n^p$, then M will have only p nonzero eigenvalues, and this initial configuration will have zero stress. If Δ is a higher-dimensional distance matrix, this configuration will represent an optimal linear projection into p dimensions.

This analysis could be carried out by fully diagonalizing M, but this is extremely wasteful when only $p \ll n$ principal eigenvectors are wanted. Instead, we use a simple iterative method based on Hotelling's power method (Hotelling, 1933). Start with a random orthonormal set of p vectors $\mathbf{e}_a$. Multiply each by the matrix M, then orthonormalize the set using the Gram-Schmidt algorithm. As this step is repeated, the vectors $\mathbf{e}_a$ approach the p largest eigenvectors.[8] The eigenvalues are then given by $\lambda_a = \mathbf{e}_a^T \cdot M \cdot \mathbf{e}_a$. Exact diagonalization is known to take $O(n^3)$ time; this method cuts the time down to $O(pn^2)$.

The second method we use for finding a good initial embedding is the **stochastic embedding** algorithm proposed by Agrafiotis and Xu (2002). This method is also very simple and fast, and seems to work well when the data is sufficiently compatible with the embedding space. The algorithm begins with a random configuration. A random edge ij is selected, and the points $\mathbf{x}_i$ and $\mathbf{x}_j$, currently separated by a distance d_{ij}, are moved along the line connecting them so their separation becomes $\alpha\Delta_{ij} + (1-\alpha)d_{ij}$. This basic step is repeated many times, while the learning rate α decreases according to a specified schedule.

2.8 Choosing the Embedding Dimension

One of the major problems with embedding algorithms is determining the intrinsic dimensionality of the data. When the dimension of the host space is increased, the optimal metric stress will always *decrease*, as the search space is enlarged. One would like to know when the embedding dimension is sufficiently large, i.e., when any additional improvement is insignificant. Principal component analysis can sometimes suggest the appropriate embedding dimension, based on the number of "large" eigenvalues of the PCA matrix M (Eq. (6)). However, in many cases the distribution of eigenvalues is relatively flat and uninformative and the subtlety then lies in setting the correct eigenvalue threshold. To our knowledge, this has not been addressed in a statistical setting. Moreover, as a linear embedding technique, PCA explores only a small subset of all possible embeddings.

We propose two complementary approaches to this question. The first method is based on a geometric analysis of the optimization problem in the space of distance matrices, and formulates the problem in probabilistic terms. It can be used to decide whether a dimensional increase is statistically significant. This method is tailored to the case of unweighted SAMMON with small distortions. The second method, on the other hand, is information-theoretic in nature, and compares embeddings based on the principle of minimum description length (MDL). This method is more heuristic than the first, and consequently more widely applicable. In the remainder of this section, we discuss both proposed methods in detail.

2.8.1 GEOMETRIC APPROACH

In practice, one often seeks the correct embedding dimension by an iterative method: successive embeddings with decreasing metric stress are constructed, in higher and higher dimensions, until

[8] Because the result will be further refined in any case, full convergence is not required.

the decrease in stress becomes negligible. We can place this iterative method on a firm statistical footing by specifying precisely what is meant by "negligible". We do this by defining a statistical null model for the decrease in stress associated with an increase in embedding dimension. For any p-dimensional embedding of a dissimilarity matrix Δ with (locally) minimum stress, our null model proposes that the remaining discrepancies between the target distances Δ_{ij} and the embedded distances are *independent* and *identically distributed* Gaussian random variables. By comparing the measured stress in a dimension $q > p$ to the stress predicted under the null model, we can assign statistical significance to the decrease in stress. When the statistical significance becomes too low, we conclude that we may well be "fitting noise," and terminate the iterative method. The details of this calculation comprise the remainder of this section.

Given a set of n points, we denote the set of all possible embeddings in p dimensions by S_n^p. The corresponding distance matrices form the manifold $D_n^p \equiv D(S_n^p) \subset \Omega_n$. This manifold is enlarged with increasing p until $p = n-1$; that is,

$$D_n^1 \subset D_n^2 \subset \cdots \subset D_n^{n-1} \equiv D_n^\infty \subset \Omega_n \ .$$

Since n points always lie in a single $(n-1)$-plane, larger values of p are never necessary. The dimension of D_n^p is the dimension of S_n^p, minus the dimension of the group of Euclidean transformations of $\mathcal{R}^p$ (i.e., the transformations $(\mathbf{X}, \hat{M})$ under which D is invariant):

$$\begin{aligned} \dim D_n^p &= \dim S_n^p - p - \frac{1}{2}p(p-1) \\ &= n \cdot p - \frac{1}{2}p(p+1) \\ &= \frac{1}{2}p\,(2n-p-1) \ . \end{aligned}$$

Equivalently, the codimension of D_n^p is

$$\begin{aligned} c_n^p &\equiv \dim \Omega_n - \dim D_n^p \\ &= \frac{1}{2}n(n-1) - \frac{1}{2}p(2n-p-1) \\ &= \frac{1}{2}(n-p)(n-p-1) \ , \end{aligned}$$

which is equal to zero when $p = n-1$, as expected.

Suppose we have found an optimal embedding $X \in S_n^p$ in p dimensions, with metric stress equal to $s(p)$. In the case of unweighted SAMMON, the stress function is simply the squared Euclidean distance between Δ and the distance matrix $D(X)$ within the encompassing space of Ω_n:

$$s(p) = ||D(X) - \Delta||^2 \ .$$

If X is a p-dimensional stress minimizer, then $D(X)$ is (locally) the closest point to Δ in D_n^p, and the **error vector** $E_p(X) \equiv \Delta - D(X)$ is perpendicular to D_n^p at that point. In other words, $E_p(X)$ lives in a space with dimension c_n^p (the codimension of D_n^p).

Assume now that we look for a q-dimensional stress minimizer ($q > p$). Starting at X, the search manifold is extended to D_n^q, adding $\dim D_n^q - \dim D_n^p = c_n^p - c_n^q$ new directions. If $E_p(X)$ is small, then a q-dimensional minimizer can be found by moving $D(X)$ so these (now unconstrained) components of $E_p(X)$ become zero. This will lead to a new error vector $E_q(X)$ with a lower stress value $s(q)$. Note that $s(q) = \sum_i E_i^2 < \sum_j E_j^2 = s(p)$, where the second sum is over all c_n^p components of $E(X)$, while the first sum is over a particular subset of c_n^q components.

At this point we ask whether the reduction in the error is significant, i.e., greater than expected by chance alone. Our null hypothesis is that the error vector is *randomly* oriented within the space perpendicular to D_n^p at $D(X)$. That is, we hypothesize that $E_p(X)$ is given by

$$\hat{E}_p(X) = \frac{(e_1, e_2, ..., e_{c_n^p})}{\sqrt{e_1^2 + e_2^2 + ..e_{c_n^p}^2}} \sqrt{s(p)} \ ,$$

where the e_i are normally distributed with zero mean and unit variance. Setting the first c_n^q coordinate axes in this space to be those that are *also* perpendicular to D_n^q, the projection of this random vector onto the subspace where $E_q(X)$ resides is

$$\hat{E}_q(X) = \frac{(e_1, e_2, ..., e_{c_n^q}, 0, 0, ..0)}{\sqrt{e_1^2 + e_2^2 + ..e_{c_n^p}^2}} \sqrt{s(p)} \ .$$

The **random stress ratio** is therefore

$$\hat{F} \equiv \frac{\hat{s}(q)}{\hat{s}(p)} = \frac{||\hat{E}_q(X)||}{||\hat{E}_p(X)||} = \frac{\sum_{i=1}^{c_n^q} e_i^2}{\sum_{j=1}^{c_n^p} e_j^2} < 1 \ .$$

This can be rewritten as

$$\hat{F} = \frac{A}{A+B} \ ,$$

where $A = \sum_{i=1}^{c_n^q} e_i^2$ and $B = \sum_{c_n^q+1}^{c_n^p} e_i^2$. Note that A and B are two independent chi-squared random variables, with a and b degrees of freedom, where

$$\begin{aligned} a = c_n^q &= \frac{1}{2}(n-q)(n-q-1) \ , \\ b = c_n^p - c_n^q &= \frac{1}{2}(q-p)(2n-p-q-1) \ . \end{aligned}$$

Given an observed stress ratio of $F = 1/(1+\varepsilon)$, we are interested in the probability that $\hat{F} \leq F$, or (equivalently) that $\varepsilon A - B < 0$. Since the distributions of A and B are known, the significance can be calculated *exactly*. However, when $p, q \ll n$, as is often the case, it is useful to approximate the significance in terms of the normal distribution, so that tabulated Z-scores may be used. Specifically, as a and b become large, A and B approach normal variables: $A \sim N(a, \sqrt{2a})$ and $B \sim N(b, \sqrt{2b})$. Therefore, the difference $\varepsilon A - B$ is distributed as

$$\begin{aligned} x = \varepsilon A - B &\sim N(\varepsilon a, \varepsilon\sqrt{2a}) - N(b, \sqrt{2b}) \\ &\sim N(\varepsilon a - b, \sqrt{2}\sqrt{\varepsilon^2 a + b}) \\ &\equiv N(\mu_x, \sigma_x) \ . \end{aligned}$$

With the scaling $z = (x - \mu_x)/\sigma_x$, the distribution is transformed to a standard normal distribution, and

$$P(\varepsilon A - B < 0) = P\left(z > \frac{\varepsilon a - b}{\sqrt{2}\sqrt{\varepsilon^2 a + b}}\right) \ ,$$

where $z \sim N(0,1)$. The probability is $1/2$ when $\varepsilon = b/a$. For the probability to be significant (say, three standard deviations away from the mean, or smaller than $P(z > 3)$), we need to have ε greater than $(b + 3\sqrt{2b})/a$.

In summary, we have derived the significance (p-value) of a given stress ratio, based on a postulated background distribution. This p-value can be calculated exactly, or approximated in terms of the normal distribution. If the p-value associated with an increase in embedding dimension is sufficiently low, then the decrease in stress is significant, and the higher-dimensional embedding describes the data (Δ) significantly better than the lower-dimensional one. In this framework, the optimal embedding dimension has been found when an increase in dimension fails to significantly decrease the metric stress.

2.8.2 Information-Theoretic Approach

An alternative method for model selection is the minimum description length (MDL) approach. The description length of a given **model (hypothesis) and data** is defined as the description length of the **model** plus the description length of the **data given the model**. In our case, we are trying to represent proximity data (Δ) in terms of the pairwise distances from a p-dimensional embedding. The model is a specific embedding $X \in S_n^p$. Given the model, the data can be reconstructed from the pairwise distortions: for concreteness, we will use the relative distortions

$$E_{ij} = \frac{D_{ij}(X) - \Delta_{ij}}{\Delta_{ij}} \ .$$

According to the MDL principle, we should select the model that minimizes the total description length. This heuristic favors low-dimensional models (short model description) that are capable of providing a fairly accurate description of the data (short description of the remaining errors, given the model).

The model is a specific embedding with the set of positions $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n$ in Euclidean space $\mathcal{R}^p$, for a total of $n \cdot p$ independent coordinates. Since the coordinates are not explicitly statistics of the data (we do not have an explicit mapping from Δ to X, but rather determine X implicitly, through optimization), it is difficult to specify the uncertainty in each coordinate, which could be used to define the description length. However, indirectly, they do summarize some global information about the data and in that sense they can be perceived as statistics. One can then estimate the uncertainty from the gradient curve in the vicinity of the point, or from the overall distortion in pairwise distances associated with it. For simplicity we assume a constant uncertainty for all coordinates, so the description length of the model is proportional to $n \cdot p$.

The description of the data given the model depends on the set of $n(n-1)/2$ pairwise distortions. Because this represents a large number of samples, the description length per sample will approach the information-theoretic lower bound, which is related to the entropy of the underlying distribution. For a continuous probability distribution $p(x)$, the entropy is $S[p] = -\int dx\, p(x) \log_2 p(x)$. According to Shannon's theorem, to encode a stream of samples from this distribution, with errors bounded by $\varepsilon/2$ (which must be small), one needs $-\log_2 \varepsilon + S[p]$ bits per sample. In our case, we must estimate the underlying distribution from the empirically measured distortions E_{ij}, which can be done along the lines of Section 2.6.1.

Combining the two terms, we suggest a scoring function of the form

$$\alpha n \cdot p + \frac{1}{2} n(n-1) S_E \ , \tag{7}$$

where S_E is the entropy of the error distribution, and the scaling parameter α represents the description length per coordinate of the model. We have dropped the constant term involving $-\log_2 \varepsilon$: since this term is independent of p and X, it plays no role when comparing different models.

Initially, we intended to train the parameter α to optimize the scoring function's performance; for instance, one might seek the α that most often assigns noisy data to its original dimension. However, upon reflection it is clear that the MDL method should not, in fact, be coerced into this behavior. The purpose of the method is to find the shortest encoding of the data, and often this will *not* coincide with the data's original dimensionality. For noisy and high-dimensional data in particular, the error distribution will never become very narrow, so the description length of the conditional data cannot become arbitrarily short, while each additional coordinate costs the same amount. Unless α is unreasonably small (say, less than 2 bits per coordinate), the MDL heuristic will select a lower dimension than it would for the denoised data. This behavior is acceptable and even informative. Therefore, we selected a somewhat arbitrary value of $\alpha = 10$ for use in Section 3.2, corresponding to a relative precision of 0.001, with the understanding that values anywhere from 5 to 50 would also be reasonable.

3. Test Data and Results

To test our algorithm we ran several tests. The first set of experiments tested the robustness and performance of different metric objective functions. The second set tested our method for determining the embedding dimension. Next we evaluate structural preservation when using our algorithm compared to MDS. Lastly, we test and compare the performance of our algorithm on handwriting data.

3.1 Comparison of Metric Objective Functions

We created sixteen random configurations of 1200 points in two and three dimensions (8 sets in 2d, 8 sets in 3d). Each configuration consisted of twelve gaussian clusters of 100 points each, with principal standard deviations between 0.2 and 1.0, and with intercluster separations between 1.0 and 8.0. A test distance matrix was generated from each configuration.

Our first experiment tested the robustness of metric MDS in the presence of noise, to see how frequently the algorithm failed to converge to the global minimum. Using our algorithm for metric SSTRESS with intermediate weighting, we embedded the test matrices 100 times each (from random initial configurations), both without noise and with multiplicative noise of strength 0.02, 0.1, or 0.5. The data sets were embedded in their original dimensions.

For the 2d→2d tests, the algorithm converged to the global minimum 100% of the time, for each test matrix and for each level of noise. For the 3d→3d tests, the algorithm found the global minimum 100% of the time for seven of the eight test matrices. On the eighth test matrix, the global minimum was found 70–80% of the time, depending on the noise; in the remaining trials, a single nonglobal minimizing configuration, with a low stress of 0.01–0.02, was found.

Our second experiment compared the performance of the various objective functions, and used the same test matrices, but truncated to 400 points. We embedded the distance matrices from 3d→3d, 100 times each (from random initial configurations), with no noise, using SSTRESS with intermediate and global weighting and SAMMON with intermediate and global weighting. The number of times each objective function converged to the global minimum is shown in Table 1. The results suggest that SAMMON is more liable to converge to a nonglobal minimizer than SSTRESS, as noted by other authors. They also indicate that global weighting, which emphasizes the importance of large target distances over small ones, is more successful than intermediate weighting at recovering the original configuration.

ID	SSTRESS-i	SSTRESS-g	SAMMON-i	SAMMON-g
1	100	100	71	100
2	100	100	100	100
3	53	100	35	34
4	87	99	25	52
5	47	100	12	13
6	49	100	38	54
7	18	82	16	25
8	100	100	39	100

Table 1: Percentage of successful trials for 3d→3d embedding, for eight 400-point test sets and four different objective functions.

3.2 Dimensionality Selection

We created twenty random configurations of 250 points in 2, 3, 5, 10, and 50 dimensions. (Four for each dimensionality: a single gaussian cluster, a closely spaced pair of clusters, a widely spaced pair of clusters, and a set of eight scattered clusters.) We then generated five dissimilarity matrices from each of these configurations, using five different metrics, for a total of one hundred test matrices. The metrics we used were:

1. Euc = Euclidean metric, $\rho(\mathbf{x},\mathbf{y}) = \sqrt{\sum_i (x_i - y_i)^2}$,
2. EucW = Euclidean metric plus weak multiplicative noise,
3. EucS = Euclidean metric plus strong multiplicative noise,
4. Mink = Minkowski metric, $\rho(\mathbf{x},\mathbf{y}) = (\sum_i |x_i - y_i|^{3/2})^{2/3}$,
5. Manh = Manhattan metric, $\rho(\mathbf{x},\mathbf{y}) = \sum_i |x_i - y_i|$.

In this set of tests, we embedded each test matrix 10 times each (from random initial configurations) in 2, 3, 4, 5, and 10 dimensions, using SAMMON with global weighting. We took the lowest stress from each set of 10 trials, and retained the corresponding embedding.

From the stresses, we calculated the statistical significance of the dimensional transitions $2 \rightarrow 3$, $3 \rightarrow 4$, $4 \rightarrow 5$, and $5 \rightarrow 10$, as described in Section 2.8.1 (geometric approach). These significances were used to determine the best embedding dimension for each data set. An increase in dimension was considered justified if it improved the stress at the 3σ-level ($P < 0.0025$, approximately). From the final embeddings, we calculated the entropy of the distribution of errors for each embedding dimension, as described in Section 2.8.2 (information-theoretic approach). Using the measured entropies, we selected the dimensionality that minimized the MDL-based scoring function given by Eq. (7).

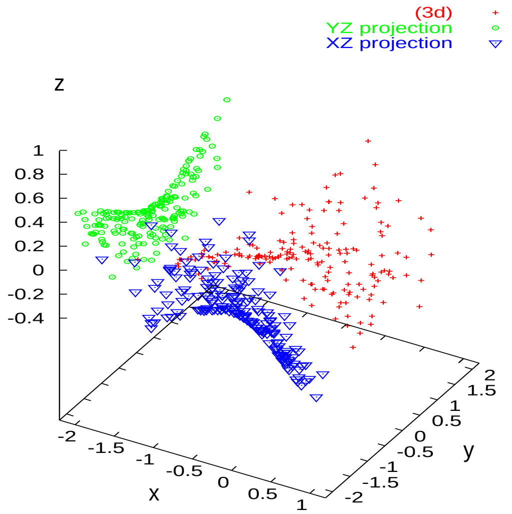

Figure 8: **Embedding a non-Euclidean space.** The dissimilarity matrix was created by applying a Minkowskian metric to a two-dimensional gaussian distribution of points. After metric embedding in 3d, the points appear to form a two-dimensional surface with negative curvature, i.e., a saddle.

Tables 2 and 3 summarize the geometric and information-theoretic results. The results were fairly consistent across the four types of test configuration (gaussian, pair, etc.). On the other hand, they depended strongly on the dimensionality of the original configuration and on the way in which dissimilarities were obtained, as seen in the Tables. Moreover, the geometric and information-theoretic approaches can lead to very different results when applied to noisy or nonmetric data.

Applying the geometric approach, our algorithm selected the original dimensionality of the data set in the Euclidean cases, both with and without noise, and indicated that higher-dimensional embeddings were significantly better at describing the Minkowski- and Manhattan-metric test sets. The results for the noisy data are not surprising; indeed, the method is designed to select the correct dimensionality for data with additive gaussian noise. For the non-Euclidean test sets, the results suggest that when a Minkowskian metric is imposed on a low-dimensional set of points, the points tend to “curl up” into a higher dimension. Visual inspection of test embeddings tends to support this idea: for instance, Figure 8 shows a 2d→3d example using the Minkowski metric in which the embedded points have formed a saddle-shaped surface.

The information-theoretic approach often proposed a *lower* embedding dimension than the geometric approach. This can best be understood by comparing the goals of the two approaches with respect to residual errors. The geometric method tries to increase the embedding dimension until the residual errors are effectively *random*, and as much information as possible has been packed into the model. On the other hand, the MDL-based method will increase the embedding dimension until a balance is struck between the residual errors and the model, such that the total description length

	Euc	EucW	EucS	Mink	Manh
$d = 2$	2	2	2	3-4	3-4
$d = 3$	3	3	3	5	5
$d = 5$	5	5	5	10	10
$d = 10$	10	10	10	10	10

Table 2: **Best embedding dimension: geometric approach.**

	Euc	EucW	EucS	Mink	Manh
$d = 2$	2	2	2	3	3
$d = 3$	3	3	3	3-4	3-4
$d = 5$	5	5	3-4	5	5
$d = 10$	10	5	3	10	10

Table 3: **Best embedding dimension: information-theoretic approach.**

is minimized. This compromise will often leave significant information in the residual errors; in any such case, the geometric approach will propose a higher dimensionality for the data.

3.3 Structural Preservation

To assess the efficacy of our distributional scaling method in preserving the structure of input data, we used the distributional method to re-embed the 100 test matrices from the previous section, in each case starting from the optimal metric embedding. For each test matrix, we calculated six different measures of structural fidelity, before and after the re-embedding. The first measure was the metric SSTRESS, which was expected to increase. The remaining measures were of the form $\sum_{A \leq B} W_{AB} D[\rho_{AB}, \tilde{\rho}_{AB}]$, where $D[p,q]$ was one of the following:

1. EMD = Earth-mover's distance,
2. JS = Jensen-Shannon distance,
3. mean = squared difference between the means of p and q,
4. max = squared difference between the maxima,
5. variance = squared difference between the variances.

Table 4 shows the percent change in each of these measures, averaged over the 100 test sets, for various embedding dimensions.

These results pertain to low-stress (metric SSTRESS ≤ 0.05) embeddings, where the agreement between distributions is rather good even without the improvements from our method. When embedded with the distributional scaling method we observe a modest increase in metric stress. However, this is compensated on average by substantial improvements in the other measures. Moreover, while we explicitly optimize only the EMD, the changes in the other measures are correlated.

For highly frustrated data, like the motivating example of Section 2.5 (shown in Figure 1, with metric SSTRESS ~ 0.5), the numbers are more dramatic. The bottom row of Table 4 shows the

Dim.	stress	EMD	JS	mean	max	variance
2	+29%	-45%	-50%	-34%	-17%	+22%
3	+27%	-59%	-61%	-44%	-24%	-24%
4	+23%	-70%	-69%	-48%	-37%	-60%
5	+20%	-76%	-74%	-47%	-45%	-83%
10	+19%	-87%	-86%	-81%	-46%	-81%
2^*	+1.7%	-49%	-60%	-79%	-17%	-22%

Table 4: **Change in six measures of structural fidelity when distributional scaling is applied.** The last data set (2^*) is the one from Figure 1.

changes in the same six measures during that example's re-embedding in $d = 2$. Here, the improvements in structural fidelity cause only a very small increase in metric stress. Our method is perhaps best suited to this type of example, where no low-stress embedding exists. In such cases, distributional MDS distinguishes among many candidate embeddings where metric MDS cannot, and selects a candidate that is faithful to the structure of the original data.

3.4 Handwriting Data

Finally, to test our algorithms on real-world data, we applied both metric and distributional SSTRESS to a subset of the MNIST database of handwritten digits.[9] Each digit is represented by a 28×28 grayscale image, where each pixel's brightness is between 0 and 255; we used the Euclidean distances between these 784-dimensional data points as input to our embedding algorithms. Figure 9 shows the two-dimensional embeddings that were generated using each method. We restrict this example to three digits, because we expect to need more than two dimensions to embed all ten digits (Saul and Roweis, 2003), making the results harder to interpret.

The general layout is similar with both methods: the digit 2 is most readily confused with the other two digits, and digits 0 and 1 are most easily distinguished from one another. However, the application of distributional scaling (right) clearly improves the embedding, in that the overlap between clusters is greatly reduced. This result suggests that the *distributions* of intercluster distances provide additional information distinguishing the handwritten digits from one another.

4. Discussion

In this paper we presented a method for structure-preserving embedding. As opposed to classical multidimensional scaling methods that are concerned only with the pairwise distances, our algorithm also monitors any higher-order structure that might exist in the data and attempts to preserve it as well.

There are many ways to characterize the structure of the data. If the data resides in a real normed space one can talk about its geometry. However, embedding is more interesting when the data is given as proximity data, where it may or may not be metric. The notion of geometry in these cases is elusive. Here we decided to focus on the clustering profile that is implied by the data. The

[9] The MNIST data is available at http://yann.lecun.org/exdb/mnist.

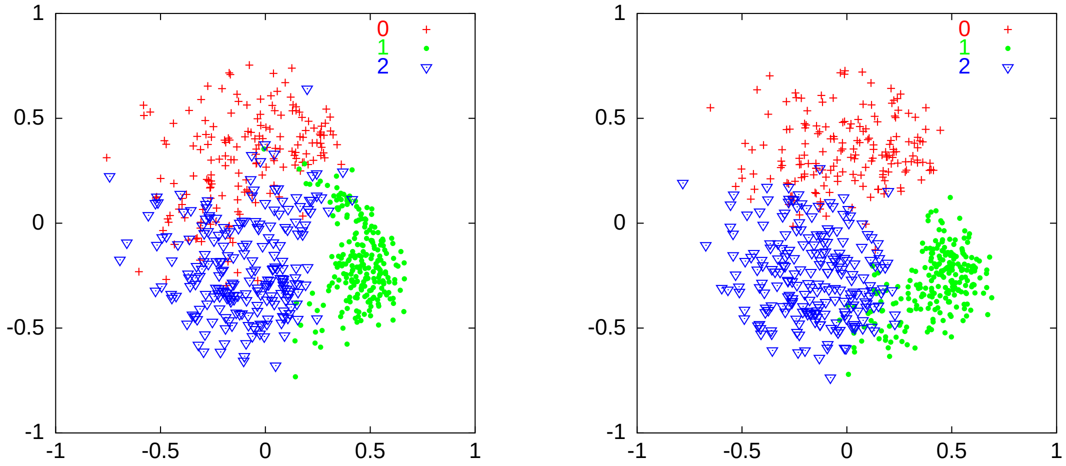

Figure 9: **Maps of handwritten digits using metric MDS (left) and distributional MDS (right).** The embeddings are of 628 examples of digits 0, 1, and 2, using the SSTRESS objective function with local weighting.

cluster structure is a strong indicator of self organization of the data and can be used to describe the structure of a variety of data types. Note that since the relative positioning of clusters with respect to each other is important in order to recover the structure, the cluster assignments alone are not sufficient.

To create embeddings that preserve the structure we defined a new objective function that considers the geometric distortion as well as the pairwise distortion. Rather than considering the error in each edge independently (as in traditional MDS techniques), we opt for embeddings that preserve the overall structure of the information contained in the matrix Δ, and specifically, the distributions of distances between and within clusters. The cluster assignments need not be known in advance, as demonstrated in Section 2.5. One can apply traditional MDS techniques to generate a preliminary embedding and use simple clustering algorithms in the host space to generate cluster assignments. Even when these assignments are imperfect, the distributional information can recover the true structure. We explored variants on this objective function, considering different functional forms, normalizations and types of dissimilarity data. Our method can be applied to proximity data as well as to high-dimensional feature vector data.

Finally, we addressed the problem of finding the “right” embedding dimension. In classical MDS techniques, the embedding dimension must be set by the user, and no bound is provided on the expected distortion of the embedding. In this paper we proposed two methods for computing the expected distortion and estimating the right dimensionality of the data: a local geometric approach, and a global heuristic based on the MDL principle.

Future directions include the study of globalization methods, other methods of assessing the structure of the data and their incorporation in the objective function, and the application of this method to real data sets.

Acknowledgments

This work is supported by the National Science Foundation under Grant No. 0133311 to Golan Yona.

Appendix A. Metric Optimization

There are numerous methods described in the literature for the numerical optimization of both the SAMMON and SSTRESS objective functions. The metric stress function is globally well-behaved: it is smooth, bounded from below, and has compact level sets. Because of this, it is easy to guarantee convergence to a local minimum. Differing strategies are distinguished not by their robustness, but by their running times, rates of convergence, and space requirements. For large data sets, evaluation of $\mathcal{H}$ takes $O(n^2)$ operations, as does the evaluation of either the gradient, $\nabla\mathcal{H}$, or the entire Hessian matrix, $\nabla^2\mathcal{H}$. As shown by Kearsley et al. (1998), linearly convergent methods, like the Guttman transform originally proposed by Sammon (1969) for SAMMON, tend to stop prematurely. On the other hand, the multidimensional Newton-Raphson method, with quadratic convergence to a local minimum, can be applied with good success. Newton's method takes $O(n^3)$ operations per iteration, most of which are spent inverting the Hessian matrix, and space of $O(n^2)$ to hold the Hessian matrix, which is not sparse. Because the latter space requirement may be prohibitive, and because we may want to check partially-converged results more frequently, we do not use Newton's method. Instead we choose a quasi-Newton minimization strategy, **conjugate gradient descent**, as an alternative.

Conjugate gradient descent shares the quadratic convergence and expected running time of Newton's method; but it has more modest storage requirements, and it produces output at shorter intervals. The theoretical basis for the method is described in many places: see, for instance, *Numerical Recipes in C* and its references (Press et al., 1993). We use the following version of the algorithm:

1. Set the iteration count k to 0, and choose an initial embedding X_0.
2. Calculate the current downhill gradient: $G_{k+1} = -\nabla\mathcal{H}(X_k)$.
3. Find new search direction, as a linear combination of the previous search direction and the gradient:
$$Y_{k+1} = G_{k+1} + \frac{(G_{k+1} - G_k)\cdot G_{k+1}}{G_k \cdot G_k} Y_k \; .$$
(For the first iteration, set $Y_1 = G_1$.)
4. Minimize $\mathcal{H}(X_k + \alpha Y_{k+1})$ with respect to the step size α. Update the embedding: $X_{k+1} = X_k + \alpha Y_{k+1}$.
5. Terminate if α, $||G_{k+1}||$, or $\mathcal{H}(X_{k+1})$ is small enough, or if k is large enough. Otherwise, increment k and return to step 2.

Because the conjugate gradient method is easy to implement and requires only first derivatives, we used it for the optimization of all the objective functions mentioned in the paper, including distributional scaling. As indicated above, it is as efficient as Newton's method and more convenient in several ways. In addition, we were able to substantially accelerate the conjugate-gradient optimization of the SSTRESS and SAMMON functions by speeding up the line minimization step, which is the bottleneck. We describe how this can be done in the following two sections.

A.1 Optimizing Metric SSTRESS

When applied to metric MDS, the conjugate gradient algorithm spends most of its time in step 4, performing line minimizations: at each iteration it calculates

$$\arg\min_{\alpha\in\mathcal{R}} \mathcal{H}(X+\alpha Y)\ ,$$

where the starting point X and the search direction Y are known. In general, pinning down each minimizing α (to sixteen digits of precision, say) will require 20–40 evaluations of $\mathcal{H}$ at different points along the ray $X+\alpha Y$. For metric SSTRESS, however, this slow process can be circumvented. Our key observation is that because $\mathcal{H}(X)$ is polynomial in the coordinates $(\mathbf{x}_i)_\mu$, its restriction to a line is also polynomial, and can be minimized in constant time once the coefficients are known. Specifically, for fixed X and Y, $\mathcal{H}(X+\alpha Y)$ is a quartic polynomial in α, with coefficients that can be found in $O(n^2)$ operations. In practice, it takes only a few times longer to find these coefficients than it does to evaluate $\mathcal{H}$ itself. As a result, by using a specialized subroutine for polynomial line minimization, we accelerate the optimization of metric SSTRESS by a factor of ten.

A.2 Optimizing Metric SAMMON

In the SAMMON case, the restriction of $\mathcal{H}$ to a line is not polynomial, so we cannot avail ourselves directly of the trick that works for SSTRESS. However, it is possible to define an auxiliary function that *is* polynomial, which can be used in place of $\mathcal{H}$ in the line minimizations. We will require the auxiliary function to be a **majorizing function** for $\mathcal{H}$; this guarantees convergence to a local minimum by ensuring that steps that decrease the auxiliary function also decrease $\mathcal{H}$.

Formally, a function $g(x,y)$ is called a majorizing function for $f(x)$ if $\forall_{x,y} g(x,y)\geq f(x)$ and $\forall_y g(y,y)=f(y)$. That is, for each fixed value of y (called the "point of support"), the values of $f(x)$ and $g(x,y)$ coincide at $x=y$, and $g(x,y)$ is never less than $f(x)$. If f and g are smooth, then clearly $\partial_1 g(y,y)=f'(y)$ and $\partial_1^2 g(y,y)\geq f''(y)$ for all y as well. Majorizing functions are of interest in minimization problems, as they give rise to the following algorithm for finding a local minimizer of f. Start at any x_0. Consider $g(x,x_0)$ as a function of x, and look for a value of x such that $g(x,x_0)<g(x_0,x_0)$. If there is none, terminate: x_0 is a (local) minimizer of f. If there is one, call it x_1. Then $f(x_1)\leq g(x_1,x_0)<g(x_0,x_0)=f(x_0)$, so we have decreased the value of f. Repeat. The potential advantage is that g can have special properties that f lacks, making it easier to minimize.

We want to find a majorizing function for $f(x;\delta)=(x-\delta)^2$ that has the additional property of being polynomial in x^2. The simplest such function is the quartic

$$g_4(x,y;\delta)=\delta^2+\left(1-\frac{3\delta}{y}\right)x^2+\frac{\delta}{y^3}x^4\ .$$

At the point of support y, only the first derivatives of g and f coincide. Using g instead of f in the conjugate gradient algorithm gives a method with first-order convergence. To maintain quadratic convergence, g needs to better approximate f for small step sizes, i.e., more derivatives need to coincide. With this constraint, the next-simplest choice is the eighth-order polynomial

$$g_8(x,y;\delta)=\delta^2+\left(1-\frac{35\delta}{8y}\right)x^2+\frac{35\delta}{8y^3}x^4-\frac{21\delta}{8y^5}x^6+\frac{5\delta}{8y^7}x^8\ .$$

This function matches f in its first three derivatives at the point of support. For fast minimization of metric SAMMON, we use the function g_8 in place of f for each line minimization.

Appendix B. Nonmetric Optimization

Nonmetric scaling is often performed by alternating between two types of steps: those that improve the configuration X, and those that improve the transformation g. Such algorithms are at best linearly convergent, since they make no use of the coupling between X and g in the objective function. Drawing on knowledge of the metric problem, we expect the nonmetric problem also to be fairly well-behaved and amenable to higher-order methods that treat X and g on the same footing. We again choose to apply conjugate gradient descent, and expect quadratic convergence to a local minimum.

In order to incorporate the function g into the set of minimization variables, we first select a parametric representation of it. For a given input matrix $\Delta = (\Delta_{ij})$, we fix $M+1$ points t_k, such that

$$t_0 < t_1 < \cdots < t_M$$

and

$$t_0 < \min_{ij} \Delta_{ij} \leq \max_{ij} \Delta_{ij} < t_M \, .$$

The t_k are chosen so that the matrix elements of Δ are distributed uniformly among the M intervals $(t_k, t_{k+1}]$. Now the function g is taken to satisfy $g(t_k) = \theta_k$ for each k, and is linearly interpolated within each interval. The requirement that g be monotonic becomes a constraint on the parameters θ:

$$\theta_0 \leq \theta_1 \leq \cdots \leq \theta_M \, . \tag{8}$$

We now minimize Eq. (2) over the range of (X, θ) admissible under the constraint (8).

Constrained minimization can be carried out in (at least) two ways consistent with our overall methodology. The first way is to employ a “simplex”-type method, analogous to that used in linear programming. Here we maintain a list of which of the M constraints $(\theta_k \leq \theta_{k+1})$ are satisfied as equalities, and take conjugate-gradient steps within that subspace. Whenever a line minimization step saturates a new inequality, we add it to the list. Whenever the downhill gradient $-\nabla\mathcal{H}$ points away from a surface $\theta_k = \theta_{k+1}$, we remove it from the list. The second way is to add a barrier function to the original objective function $\mathcal{H}(X,\theta)$. Specifically, we might minimize

$$\mathcal{H}^*(X,\theta;\mu) = \mathcal{H}(X,\theta) - \mu \sum_{k=0}^{M-1} \log\left(\frac{\theta_{k+1}-\theta_k}{\theta_M - \theta_0}\right)$$

for a sequence of barrier heights μ tending to zero. This barrier function, like $\mathcal{H}$ itself, is chosen to be scale-invariant.

Whichever method we use to enforce the constraints, we can still take advantage of efficient line minimization in the case of SSTRESS. Because of our parametrization of g, each $g(\Delta_{ij})$ is a linear function of θ; so the numerator and the denominator of Eq. (2) are polynomial in the coordinates $\mathbf{x}_{i,\mu}$ and the parameters θ_k. The restriction of $\mathcal{H}$ to a ray is

$$\mathcal{H}(X+\alpha Y, \theta+\alpha\zeta) = \frac{P_1(\alpha)}{P_2(\alpha)} \, ,$$

where P_1 and P_2 are polynomials (quartic and quadratic, in this case) with coefficients we can calculate relatively quickly. As long as the number of intervals M is small compared to n^2, evaluating the barrier function for multiple values of α will not contribute substantially to the time. However, we do not have a corresponding shortcut for nonmetric SAMMON. Therefore, our implementation of nonmetric SSTRESS is from five to ten times faster per iteration than nonmetric SAMMON.

References

D. K. Agrafiotis and H. Xu. A self-organizing principle for learning nonlinear manifolds. *Proceedings of the National Academy of Arts and Sciences*, 99:15869–15872, 2002.

I. Apostol and W. Szpankowski. Indexing and mapping of proteins using a modified nonlinear Sammon projection. *Journal of Computational Chemistry*, 20:1049–1059, 1999.

W. Basalaj. Incremental multidimensional scaling method for database visualization. In *Visual Data Exploration and Analysis VI (Proceedings of the SPIE)*, volume 3643, pages 149–158, 1999.

M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral eigenmaps for embedding and clustering. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, volume 14, pages 585–591. The MIT Press, 2002.

M. Blatt, S. Wiseman, and E. Domani. Data clustering using a model granular magnet. *Neural Computation*, 9:1805–1842, 1997.

T. F. Cox and M. A. A. Cox. *Multidimensional Scaling*. Chapman and Hall CRC, second edition, 2001.

D. L. Donoho and C. Grimes. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Arts and Sciences*, 100:5591–5596, 2003.

S. Dubnov, R. El-Yaniv, Y. Gdalyahu, E. Schneidman, N. Tishby, and G. Yona. A new nonparametric pairwise clustering algorithm based on iterative estimation of distance profiles. *Machine Learning*, 47:35–61, 2002.

R. El-Yaniv, S. Fine, and N. Tishby. Agnostic classification of Markovian sequences. In M. I. Jordan, M. J. Kearns, and S. A. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10, pages 465–471. The MIT Press, 1998.

Y. Gdalyahu, D. Weinshall, and M. Werman. A randomized algorithm for pairwise clustering. In M. S. Kearns, S. A. Solla, and D. A. Cohn, editors, *Advances in Neural Information Processing Systems*, volume 11, pages 424–430. The MIT Press, 1999.

J. C. Gower. Some distance properties of latent root and vector methods in multivariate analysis. *Biometrika*, 53:325–338, 1966.

H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441,498–520, 1933.

A. J. Kearsley, R. A. Tapia, and M. W. Trosset. The solution of the metric STRESS and SSTRESS problems in multidimensional scaling using Newton's method. *Computational Statistics*, 13: 369–396, 1998.

R. W. Klein and R. C. Dubes. Experiments in projection and clustering by simulated annealing. *Pattern Recognition*, 22:213–220, 1989.

H. Klock and J. M. Buhmann. Multidimensional scaling by deterministic annealing. In *Proceedings of the International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 245–260, 1997.

S. Kullback. *Information Theory and Statistics*. John Wiley and Sons, 1959.

E. Levina and P. Bickel. The earth mover's distance is the Mallows distance: Some insights from statistics. In *Proceedings of the Eighth IEEE International Conference on Computer Vision*, pages 251–256, 2001.

J. Lin. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, 1991.

N. Linial, E. London, and Yu. Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15:215–245, 1995.

S. W. Malone and M. W. Trosset. A study of the stationary configurations of the SSTRESS criterion for metric multidimensional scaling. Technical Report 00-06, Department of Computational & Applied Mathematics, Rice University, 2000.

W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, second edition, 1993.

V. Roth, J. Laub, M. Kawanabe, and J. M. Buhmann. Optimal cluster preserving embedding of non-metric proximity data. Technical Report IAI-TR-2002-5, University of Bonn, Informatik III, 2002.

S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.

Y. Rubner, C. Tomasi, and L. B. Guibas. A metric for distributions with applications to image databases. In *Proceedings of the Sixth IEEE International Conference on Computer Vision*, pages 59–66, 1998.

J. W. Sammon. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, 18:401–409, 1969.

L. K. Saul and S. T. Roweis. Think globally, fit locally: Unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research*, 4:119–155, 2003.

J. Shi and J. Malik. Normalized cuts and image segmentation. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 731–737, 1997.

P. S. Smith. Threshold validity for mutual neighborhood clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 15:89–92, 1993.

J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.

Z. Wu and R. Leahy. An optimal graph theoretic approach to data clustering: theory and its application to image segmentation. *PAMI*, 15:1101–1113, 1993.

G. Yona. *Methods for Global Organization of the Protein Sequence Space*. PhD thesis, The Hebrew University, Jerusalem, Israel, 1999.

G. Young and A. S. Householder. Discussion of a set of points in terms of their mutual distances. *Psychometrika*, 3:19–22, 1938.