Nonlinear Mapping: Approaches Based on Optimizing an Index of Continuity and Applying Classical Metric MDS to Revised Distances

Ulas Akkucuk
J. Douglas Carroll
Rutgers Business School – Newark and New Brunswick

Abstract

Dimensionality reduction techniques are used for representing higher dimensional data by a more meaningful lower dimensional structure. In this paper we will study two such approaches, namely Carroll's Parametric Mapping (to be abbreviated PARAMAP) (Shepard & Carroll, 1966) and a variation of ISOMAP (Tenenbaum, de Silva, & Langford, 2000). The former relies on iterative minimization of a cost function while the latter applies classical MDS after a preprocessing step involving the use of a shortest path algorithm to define approximate geodesic distances. We test the PARAMAP algorithm extensively on the 62 points on the surface of a sphere with varying error levels and degrees of irregularity of spacing as well as well as the Swiss roll data which was initially used to test ISOMAP. We also compare the results to ISOMAP produced embeddings of the sphere and the Swiss roll. We develop a measure of congruence between the input data and the mapped low dimensional embedding based on the preservation of the local structure, and compare the different approaches using this measure. This measure can also be used to test the statistical significance of an obtained configuration with the help of a randomization procedure we developed and applying simple probabilistic concepts (Chebyshev's inequality). The PARAMAP approach turns out to be very superior compared to the ISOMAP approach in both kinds of data we tested the programs on. Especially in the case of the sphere, ISOMAP produces mappings resemble a projection of the points on the lower dimensional space, much like what the essentially linear classical MDS or PCA analysis would also do. PARAMAP on the other hand manages to preserve the local structure much better than ISOMAP does, even in the case of the Swiss roll data which was used in the original paper to show the effectiveness of ISOMAP (Tenenbaum et al., 2000). The difference in the mappings of the two algorithms is much more severe in the case of the sphere (which is a "closed" data set) as opposed to the Swiss roll (which is an "open" data set). This suggests that ISOMAP in its current state would have limited use in such "open" or "semi-closed" data sets. In "closed" data sets, however, PARAMAP seems to be the only option. However, PARAMAP has its shortcomings in the sense that it requires extensive computer run time. Computational improvements in PARAMAP are a must if it needs to be generalized to a much larger number of points (say 1,000 or more). Also ISOMAP and PARAMAP both could benefit from a procedure that would judiciously select a subset of points to be analyzed and then, after analysis, extrapolate or interpolate the rest. Our current research direction, is to work on improving computational efficiency of PARAMAP and also come up with an efficient extrapolation or interpolation procedure.

Introduction

A general problem in multivariate analysis is to determine a smaller set of variables necessary to account for a larger number of observed variables. Principal-components analysis (PCA) and multidimensional scaling (MDS) are two ways to deal with this problem. However, they both fail to recover the true dimensionality unless the relationship between the observed variables and the smaller set of underlying variables is linear or near linear (Mardia, Kent, & Bibby, 1979). Alternative approaches are needed when this relationship is highly nonlinear.

Shepard and Carroll (1966) presented two such mapping approaches. One of the methods, proposed by Shepard, used locally monotone analysis of proximities. This method was essentially an application of nonmetric two-way multidimensional scaling (MDS) analysis, in which only the small distances were used as input, with the large distances treated as “missing”, so that the distances in the mapping produced were only approximately locally monotonically related to the derived distances defined via the variables provided as input to the procedure. In a variation, metric MDS could be used, in which case the mapping tends to preserve a locally linear relation between data distances and recovered distances. A similar approach was also used by Buja and Swayne (2002) in a method they have called “within-groups MDS”. The other approach, proposed by Carroll, reduced the dimensionality by optimizing an index of “continuity” or “smoothness”. The former worked well if the nonlinearities were not too severe; in particular as long as the inverse mapping from the underlying “latent (lower dimensional) space” to the space defined by the observed variables does *not* comprise a closed (or near closed) curve (such as a circle) or surface (such as a sphere or hypersphere). The latter approach did not have such a restriction. The latter approach was also incorporated into a computer program called PARAMAP standing for “PARAmetric MAPping of nonlinear data structures” (Carroll & Chang, 1968; Carroll, 1968). Parametric mapping was intended to find a mapping of points on a nonlinear surface embedded in a high-dimensional vector space, whose topological dimensionality is smaller than that of the embedding space, to a lower-dimensional vector space, whose dimensionality equals that topological dimensionality, and which preserves that topology, with the possible exception of a single point (the point where the surface has to be “punctured” to effect the mapping) at which a severe discontinuity in the mapping will necessarily occur. (If the surface is not closed no such discontinuity would occur.)

The PARAMAP computer program was tested on several data sets including 20 equally spaced points on a circle (where the topological dimensionality is one), 62 regularly spaced points on a sphere (where the topological dimensionality is two), and 49 points equally spaced on a very specially constructed four-dimensional torus (where the topological dimensionality is also two). The results were all successful for the errorless configurations described above. The PARAMAP analysis of the 20 points on the circle in effect cut the circle between some pair of adjacent points, and then mapped the 20 points onto a line, with equal spacing on this line, and with the order of the points maintained except, of course, for the point at which the circle is “cut” (Figure 1 shows the plot of original coordinates y_1 and y_2 against the recovered coordinate x). In the case of the 62 points on the sphere, a mapping (Figure 2a) was obtained which resembled very closely one of the fairly standard cartographic mappings (one called an “azimuthal equidistant projection” such as that shown in Figure 2b) of the globe (Deetz & Adams, 1945). In the case of the azimuthal equidistant projection, the sphere is in fact “cut open” or “punctured” at an arbitrary point along the equatorial parallel, where the points are less densely distributed than at other points, resulting in a minimal distortion of distances almost everywhere on the resulting mapping. At the point on the equatorial parallel where this “puncture” occurs though, the distances are very severely distorted. In the case of the 49 equally spaced points on the highly regular torus embedded in four dimensions, the mapping produced an almost perfect two-dimensional lattice as shown in Figure 3 (one of 49 equally likely lattices that could have resulted).

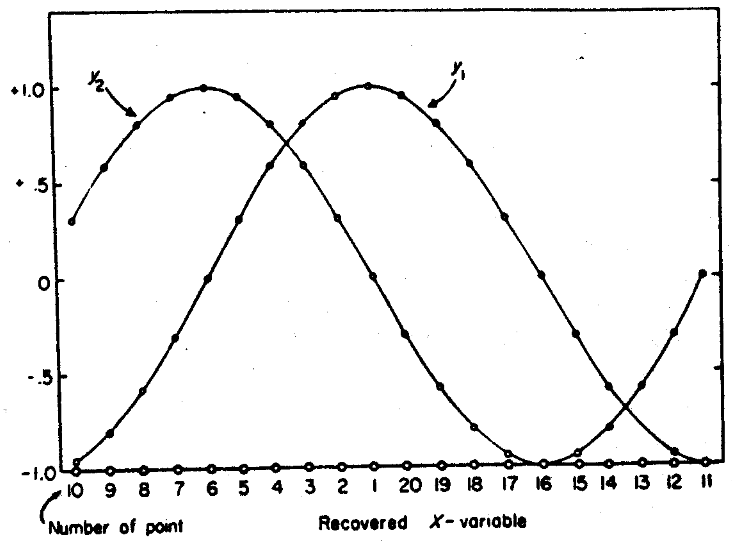

Figure1
One-dimensional solution for 20 points equally spaced around a circle (line at bottom), with original coordinates plotted above [reproduced from Shepard, & Carroll (1966)].

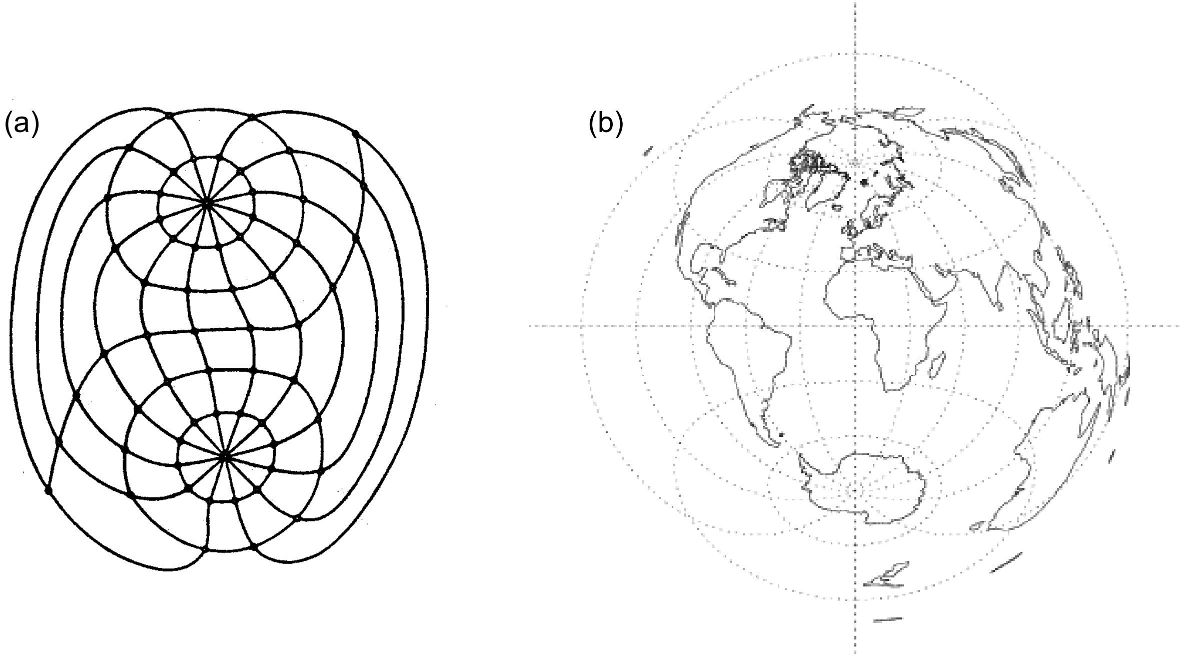

Figure 2
(a) The solution of the 62 regularly spaced and errorless points on the sphere, and (b) the azimuthal equidistant projection of the globe [(a) reproduced from Shepard and Carroll (1966) and (b) taken from http://mathworld.wolfram.com].

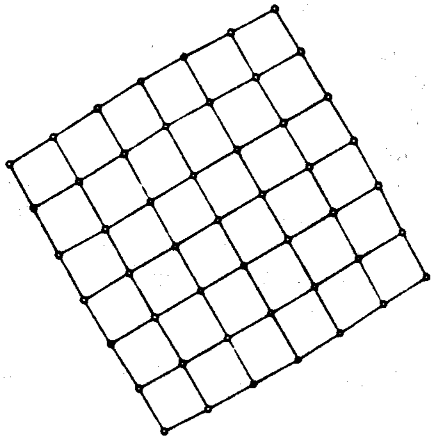

Figure 3
Two-dimensional solution for the 49 points on the surface of a torus (with analogues of the original intersecting parallels drawn in). [Reproduced from Shepard & Carroll (1966)]

Although results were successful, there was a severe local minimum problem. A large number of runs were needed to obtain a decent solution, while no documentation can be found as to the exact amount. This problem got more severe when error was added to the regularly spaced points, or when the spacing was made less regular (or both). The predicament of having to make many computer runs with less developed computing facilities stalled the development of the program at that time.

Our current work was stimulated in part by the publication of two articles on the issue of nonlinear mapping (Tenenbaum, de Silva, & Langford, 2000; Roweis & Saul, 2000). The former paper describes an approach called ISOMAP (ISOmetric MAPping), which applies classical MDS after reconstructing a distance matrix that reflects the true geodesic distances rather than the straight line Euclidean distances. The latter works by means of a method called LLE (locally linear embedding) which is very similar to the locally monotone MDS approach proposed by Shepard as used in Shepard and Carroll (1966) explained above, and also appears to be closely related to a procedure experimented with by Kruskal (1975). Both these methods deal easily with some subset of data structures that we can refer to as "open" structures, but fail to recover the underlying configuration when the data structure is "closed" such as that of the regularly arranged points on a sphere. It turns out Carroll had tried a similar approach to ISOMAP using a different shortest path algorithm (Hu, 1967) but had abandoned the work because of unsatisfactory results, particularly in attempts to generalize this approach to work for the "closed" surfaces such as points on the sphere.

Our work will focus primarily on evaluating the efficiency of the PARAMAP algorithm on the regularly and irregularly spaced points on the sphere with varying error levels. We will also test how ISOMAP works when such closed surfaces are used as input. In order to compare resulting configurations from different algorithms, we will develop a quantitative measure to assess how well local structure is preserved in the output configuration. Rotational indeterminacy tends to be a problem when the results from different runs need to be compared. In order to overcome this problem we will use standard software that performs ordinary least squares fit.

The paper will proceed as follows: In the next section we will review the important algorithmic properties of the two nonlinear mapping approaches ISOMAP and PARAMAP, and also give a cursory overview of some other approaches. Afterward, we will present our experimental procedures and explain the proposed measure for determining the congruence of the mapped solution to the input data. We will then present our results and conclusions.

Nonlinear Mapping Algorithms

We should first define the general terminology before we go on to describe the particular algorithms. We denote by:
n : number of objects
M : dimensionality of the input coordinates, i.e., the configuration for which we would like to find an underlying lower-dimensional embedding, which might also be referred to as the matrix of criterion variables, or as the criterion (multivariate) variable.
R : dimensionality of the space of recovered configuration, where $R<M$
$\mathbf{Y}$: $n \times M$ input matrix
$\mathbf{X}$: $n \times R$ output matrix
The distances between point i and point j in the input and output spaces respectively are calculated as:

$$\delta_{ij}^2 \equiv \sum_{m=1}^{M} (y_{im} - y_{jm})^2, \qquad \forall\ i, j$$

$$d_{ij}^2 \equiv \sum_{r=1}^{R} (x_{ir} - x_{jr})^2, \qquad \forall\ i, j$$

$\Delta \equiv [\ \delta_{ij}\ ]$, and $\mathbf{D} \equiv [\ d_{ij}\ ]$.

Parametric Mapping Approach

Parametric mapping approach works via optimizing a function of "continuity" or "smoothness" that relates the Euclidean distances in the output configuration to the dissimilarities coming from the input configuration. The underlying idea is based on the assumption that a continuous function (possibly perturbed by error) relates the input variable to the "underlying latent variables" to be solved for or recovered as output variables (of lower dimensionality). The simplest case is illustrated by considering discrete values of a criterion variable denoted by y_i corresponding to the ordered and equally spaced values of a predictor or input variable denoted by x_i. In this case, we would say that y values change in a reasonably "continuous" or "smooth" manner as we move along x, if changes in y as we move from the previous value of x to the next are small compared to the changes in y that occur for larger changes in x. Such an index of "smoothness" or "continuity" was already defined in the context of time series data (von Neuman, Kent, Bellison, & Hart, 1941; von Neuman, 1941). This index is given as follows:

$$\frac{\delta^2}{S^2} = \frac{1}{n-1} \sum_{i=1}^{n-1} (y_{i+1} - y_i)^2 \bigg/ \frac{1}{n} \sum_{i=1}^{n} (y_i - \overline{y})^2$$

If indeed the x values are not equally spaced the numerator of the above equation can be modified such that the changes in y are normalized by dividing by the changes in x. In other words we can represent the numerator as:

$$\delta^2 = \frac{1}{n-1}\sum_{i=1}^{n-1}\left(\frac{\Delta y_i}{\Delta x_i}\right)^2$$

The best approximations of Δy and Δx in the multidimensional case are the respective Euclidean distances. To generalize this measure to the multidimensional case one more modification is needed. When points are mostly irregularly spaced, it is necessary to look at all *(i, j)* pairs rather than just adjacent ones. So we should sum the term $(\Delta y/\Delta x)^2$ not just for adjacent points, but for all pairs of points, since "adjacency" is not well defined for the multidimensional case. To place more weight on the ratios where Δx is small we also need to multiply by a constant that is monotonically decreasing with the separation Δx. Finally, a normalization factor is needed that will perform the job S^2 did for the unidimensional case. By following this rationale, Shepard and Carroll reached the measure they called κ[1]:

$$\kappa = \sum_{i \neq j}\sum \frac{\delta_{ij}^2}{d_{ij}^4} \bigg/ \left[\sum_{i \neq j}\sum \frac{1}{d_{ij}^2}\right]^2$$

This index measures the "continuity" inversely. It needs to be minimized for a given input data configuration **Y**. The minimization can be achieved by using a steepest descent method starting with an arbitrary starting configuration for **X**. When programming the minimization procedure, there are certain changes made to this function. First $(d^2_{ij} + Ce^2)$ is substituted for d^2_{ij} wherever it appears in the equation. The variable e takes on values between 0 and 1, and has an effect on the speed of the minimization algorithm. The constant C is equal to 2 / (n - 1) and makes the variable e^2 directly interpretable as a proportion of variance. For a good starting configuration the value of e should be close to 0 say 0.05. For a bad starting configuration the value should be higher; say 0.95. As the minimization proceeds, the value of e should converge to a value close to 0. There are also two more constants that are multiplied by κ but do not affect the minimization: In the numerator the constant z, and in the denominator $[2/n(n-1)]^2$. The latter is simply another normalizing constant while the former serves to make κ attain a lowest possible value of 1. The lowest possible value of the objective function is achieved if a perfect mapping has been obtained. The value of z therefore is the inverse of the value that κ would attain if, for all i and j, $d_{ij} = \delta_{ij}$ (or more generally $d_{ij} \propto \delta_{ij}$). The function to be minimized via the steepest descent method thus becomes:

$$\kappa = z\sum_{i<j}\frac{\delta_{ij}^2}{d_{ij}^4} \bigg/ \left[\frac{2}{n(n-1)}\sum_{i<j}\frac{1}{d_{ij}^2}\right]^2,$$

with $$z = \frac{1}{4}[n(n-1)]^2\left[\sum_{i<j}\frac{1}{\delta_{ij}^2}\right].$$

We implemented this algorithm in a computer program written in C++ and compiled under UNIX environment by GNU-GCC compiler (Version 2.95.3 of 2001). Since entrapment in

[1] κ as defined here is a special case of a more general index defined in Shepard and Carroll (1966). The full rationale for this general measure, and the particular implementation we use here, is best explained in that earlier paper. One condition κ (and its generalizations) were all required to satisfy, called "condition of similitude", stipulated that the minimum possible value of κ (and generalizations) would occur when the recovered configuration was related to the input configuration via a *similarity* transformation for which Euclidean distances are either preserved or simply multiplied by a positive constant.

a local minimum is one of the biggest problems the program faces, we designed it such that it will run for a specified number of random starts specified by the user. In this case the program will save the best configuration and the κ value associated with this solution. The user will also specify the maximum number of iterations, starting value of the variable *e,* and the dimensionality *R* to be recovered. If number of random starts is given as 0, the program will expect a starting configuration to be supplied by the user. In order to save time the initial minimization procedure using random starting configurations stops each run within the session when the maximum number of iterations is reached, not when a strict convergence criterion has been met. Using the feature of supplying an initial configuration, the best solution obtained from running the program for a given number of random starts can be further fine-tuned, by permitting a much larger number of iterations. We found that this post-processing step results in superior configurations although not necessarily resulting in significantly reduced κ values.

ISOMAP Approach

The basic idea behind ISOMAP[2] (Tenenbaum et. al., 2000) is to overcome the difficulties of MDS by defining a new metric and substituting that for the Euclidean distance. We can see why the Euclidean metric is not a correct interpoint dissimilarity measure for some data structures in the example given in Figure 4 (Lee, Landasse, & Verleysen, 2002). The spiral is seemingly two-dimensional but the underlying dimensionality is one. If classical MDS is applied directly to the Euclidean distance matrix, what is most likely to happen is a projection of the points in different layers of the spiral onto each other. The spiral will not be "rolled out" but will be "compressed". Therefore, unfolding the spiral into a straight line will not be possible. Another example could be seen as the distance one has to travel when visiting a city at exactly the opposite end of the globe – a good approximation to this would be Shanghai, PRC and Buenos Aires, Argentina. The distance will be approximately 20,000 km, about half the circumference of the earth, not 12,700 km, which is the Euclidean distance (the diameter of the earth). The ISOMAP algorithm thus aims at first predicting the best approximation of this so-called "geodesic" distance as it is shown in Figure 4c or as demonstrated by the global intercity distance example, before applying the classical MDS procedure due to Torgerson (1958).

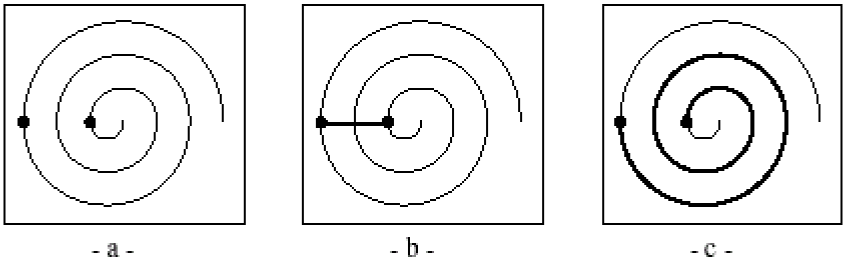

Figure 4
(a) two points in a spiral, (b) the Euclidean distance between the same two points and (c) the curvilinear or geodesic distance (Lee et al. 2002).

To obtain the approximate geodesic distances, ISOMAP implements a neighborhood graph that links or connects the closer points, for which the Euclidean distances closely

[2] ISOMAP is implemented in MATLAB and the MATLAB code, data sets and other supplementary material are available at the web site http://isomap.stanford.edu

approximate the geodesic ones, but does not connect the more distant points as measured by the Euclidean metric in the embedding space. Since only closer points are now linked by an arc, the more distant points can be reached from each other by traversing the network of nearby connected points. Taking as input matrix **Y** (configuration of n points in M dimensions) ISOMAP produces the output matrix **X** (configuration in R dimensions) taking the following steps:

1. Randomly select l landmark points. This step is optional and reduces computational time radically if n is very large. However, if this optional step is taken the program does not interpolate or extrapolate the remaining points.

2. Compute the n by n (assuming step one is skipped) input distance matrix $\Delta\equiv[\delta_{ij}]$. Using a user supplied parameter k or ε, modify Δ such that the distances that do not fall within either k nearest neighbors of point i or that are further away than the specified ε criteria are marked as infinity. That is the very distant points are disconnected, so that they will only be reached by following the shortest path involving nearby points. Let Δ_G be the resulting matrix.

3. Apply a shortest path procedure to Δ_G. The program has two options for doing so. One is Floyd's algorithm (Floyd, 1962) the other is Dijkstra's algorithm (Dijkstra, 1959). The standard option is Floyd's and proceeds as follows: For each k ranging from 1 to n we redefine $\delta_G(i,j)$ for each i and j=1,…,n ($i<j$) by $\min\{\delta_G(i,j), \delta_G(i,k) + \delta_G(k,j)\}$. Now Δ_G contains the shortest path distances from point i to j.

4. Construct the R-dimensional embedding by applying classical MDS to the matrix of distances obtained after Step 3.

Carroll, had tried a similar approach, but using Hu's (1967) shortest path algorithm. Indeed the three shortest path algorithms Floyd's, Dijkstra's, and Hu's are very similar and all scale as $O(N^3)$. However, Floyd's is the fastest running algorithm unless sparsity in the graph structure is exploited. Empirical studies suggest Dijkstra's algorithm runs on the average 1.6 times slower than Floyd's (Foster, 1995). This result is because Dijkstra's algorithm keeps an adjacency matrix along with making the computations. Our own studies suggest Hu's algorithm runs about 2.8 times slower than Floyd's does. This is because a step very similar to Floyd's algorithm is performed two times, one forward (meaning the indices $i, j,$ and k are incremented from 1 to n) and one slightly more costly backward step (meaning the indices $i, j,$ and k are decremented from n to 1). Hu also later came up with decomposition type algorithms that made use of sparsity in the graph (1968, 1969). The version of Dijkstra used in the ISOMAP program based on Kumar, Grama, Gupta, and Karypsis (1994) also makes use of sparsity and is advised when n is higher than 1,000.

The strong aspect of this program compared to PARAMAP is its noniterative nature. Therefore run times even for very large matrices are reasonable. PARAMAP cannot match this aspect because of the severe local minima problem. However, our studies have also shown that ISOMAP produces reasonable results only in a special subset of nonlinear manifolds, i.e. those which are not closed such as an open box, open cylinder, or Swiss roll (as the authors call the data set they used, which is merely a three-dimensional generalization of the spiral in Figure 4). For closed data sets such as the sphere, torus or even a closed cylinder, ISOMAP will produce only a projection of the points in which many distant points are projected very close to or even on top of each other, since ISOMAP is not able to puncture and, so to speak, "pry open" the closed manifold (or sometimes to open the surface by even more complicated "cuts"; e.g., the kind of complex pair of cuts required to open the four-dimensional torus to form the regular lattice of 49 points, as illustrated in Figure 14 of Shepard & Carroll 1966).

Other Approaches

There are several other well-known techniques that could be used for dimensionality reduction. We will describe a few of those without getting into detail.

Nonmetric multidimensional scaling. This method is ordinarily intended for uncovering the underlying stimulus configuration when the available data are in the form of dissimilarity or similarity judgments that are ordinal in nature (Kruskal, 1964a; 1964b). However, if one computes the distances from the input configuration and treats them as dissimilarities, it can also be used to extract a lower-dimensional embedding. Nonmetric two-way MDS seeks to minimize the "badness of fit" function called STRESS (in one of two variants called STRESS1 and STRESS2; Kruskal & Carroll, 1969) that relates the distances calculated from the current configuration to the pseudo-distances derived via an OLS monotone regression function (labeled $\hat{d}_{ij}$).

$$STRESS1 = \sqrt{\frac{\sum_{i<j}\left(d_{ij} - \hat{d}_{ij}\right)^2}{\sum_{i<j} d_{ij}^2}} \text{ or } STRESS2 = \sqrt{\frac{\sum_{i<j}\left(d_{ij} - \hat{d}_{ij}\right)^2}{\sum_{i<j}(d_{ij} - \bar{d}\ )^2}}$$

$$\text{where } \bar{d} = \frac{2}{n(n-1)}\sum_{i<j}^{n} d_{ij} \quad \text{and} \quad n = \text{number of objects}$$

It should also be noted that the use of nonmetric MDS with missing data is needed to implement the "locally monotone MDS" approach of Shepard's (Shepard & Carroll, 1966). If some data are missing, or treated as missing, the summations indicated are all over non-missing values of *i* and *j*. (Where $\bar{d}$ is defined by dividing the distances by the number of non-missing values.)

Sammon's Mapping. Minimizes a mapping error function that relates the distances in the obtained configuration to the dissimilarities in the input configuration (Sammon, 1969). The only important difference between Sammon's mapping and nonmetric MDS is the lack of a monotone regression function. But indeed they are so similar, Kruskal (1971) indicated that the proper options used with MDSCAL 5M (early nonmetric MDS software) would give exactly the same results[3]. The same is true of the later KYST family of algorithms (Kruskal, Young, & Seery, 1977). A comparison made among several mapping algorithms found that the two mapping algorithms produced about equally effective results based on a criterion of mapping performance (De Backer, Naud, & Scheunders, 1998). The function minimized by Sammon's mapping is given below.

$$E = \frac{1}{\sum_{i<j}\delta_{ij}}\sum_{i<j}\frac{[\delta_{ij} - d_{ij}]^2}{\delta_{ij}}$$

Others. Multidimensional scaling by iterative majorization using radial basis functions has been proposed by Webb (1995). He indeed applied his algorithm to 500 points uniformly distributed over a part sphere with added noise. The "part" sphere's surface covered the lower hemisphere and the upper hemisphere up to a latitude of 45 degrees. The resulting plot is given in Figure 5. This shows the sphere barely "opened out" as we can see. There is no indication as

[3] Kruskal's (1971) argument involved using a weighted variant of the STRESS1 Function, with weights defined in a particular way as an inverse function of the input data.

to how the algorithm would act had the sphere been closed, but we suspect strongly that it would *not* have preserved the topology, but would have effectively projected some distinct points on top of each other. Another algorithm is the CDA (Curvilinear Distance Analysis) approach by Lee et al. (2002), which is indeed a variation of ISOMAP. Other well known approaches include Self-organizing Map (SOM) (Kohonen 1990, 1995) and Auto-associative Feedforward Neural Networks (AFN) (Baldi & Hornik, 1989; Kramer, 1991). According to De Backer et al. (1998) SOM and AFN result in more successful mappings (when compared to nonmetric MDS and Sammon's mapping based on a measure of classification performance involving *k*-nearest neighbors classification rate) but are computationally more expensive. Specifically AFN might take 10 to 100 times more computing time than nonmetric MDS depending on the data set being used. SOM may occasionally run faster than MDS; however SOM can run much slower with higher *m*, a dependence that nonmetric MDS does not have (de Backer et al., 1998).

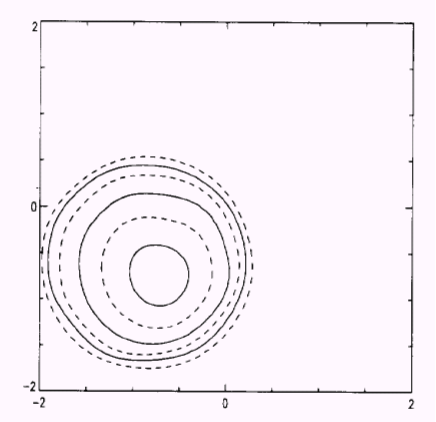

Figure 5
Projection of a ¾ sphere by iterative majorization using radial basis functions. The contours represent lines of latitude from $I\pi/10$, I=-4, -3, -2, -1, 0, 1. The outer solid line is a projection of the points in the equator while the outermost dashed line is a projection of the parallel where the sphere has been cut off (Webb, 1995).

Experimental Design and Methods

In our study to date our primary focus has been the sphere with 62 points. The base level errorless sphere has 62 points in three-dimensional Cartesian space located at the intersections of 5 equally spaced parallels and 12 equally spaced meridians. We generated the coordinates as follows:

$$x = \cos(\theta)\sin(\phi)$$

$$y = \cos(\theta)\cos(\phi)$$

$$z = \sin(\theta)$$

To generate the 60 points other than the two poles, we cross the following five values of θ (all in degrees) defining the five equally spaced parallels:
-60, -30, 0, 30, 60
with the following 12 values of ϕ (all in degrees) defining the 12 equally spaced meridia:
-150, -120, -90, -60, -30, 0, 30, 60, 90, 120, 150, 180
(Note: ϕ = 180° is equivalent to ϕ = -180°),
to obtain the 5 × 12 = 60 points excluding the two poles.

Finally the two poles are given by the coordinates:
$x = 0$, y = 0, z = -1
$x = 0$, y = 0, z = 1
which are labeled points 1 and 62 respectively. The rest of the points are numbered in a similar way between 2 and 61. For example 2 is the point that corresponds to $\theta = -60$ and $\phi = -150$. Numbering the points this way enables us to interpret the resulting two-dimensional configurations much more clearly way. A plot of the sphere (or, more precisely, of the polyhedron embedded in the sphere whose vertices comprise the 62 points with which we deal) in three dimensions can be seen in Figure 6.

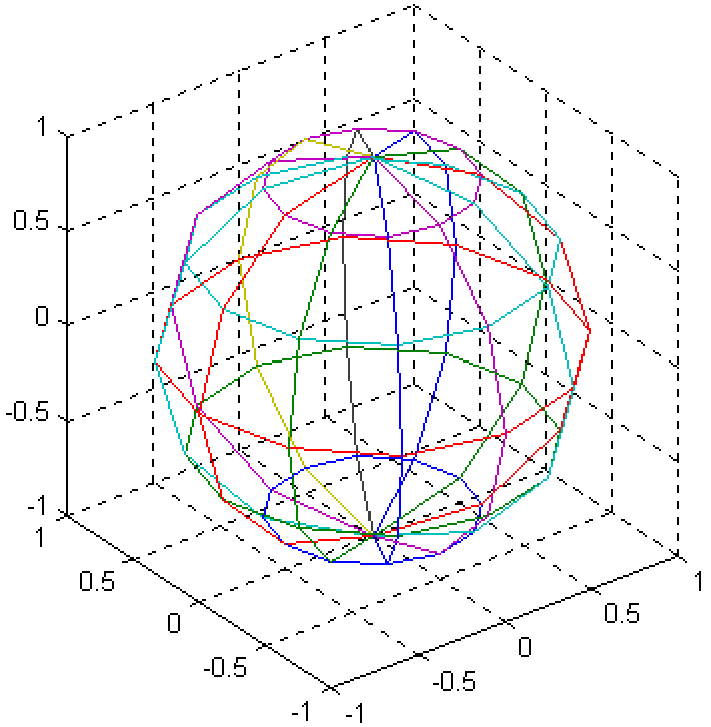

Figure 6
The sphere with 62 regularly spaced points.

Error Levels

To test PARAMAP under more realistic circumstances, we also decided to add various levels and types of error to the regular data. First we assign three levels of what we call Type A error defined as follows, to the *x*, *y*, and *z* coordinates of point *i*:

$$x_i' = x_i + s_x \cdot \sqrt{p_j} \cdot \alpha(x_i)$$

$$y_i' = y_i + s_y \cdot \sqrt{p_j} \cdot \alpha(y_i)$$

$$z_i' = z_i + s_z \cdot \sqrt{p_j} \cdot \alpha(z_i)$$

where s_x , s_y , s_z are the standard deviations of the coordinates respectively, p_j is a factor that determines the error expressed as a percentage of the variance for error level j, $\alpha(v_i) \sim$ i.i.d. $N(0,1)$ where $v_i = x_i, y_i$, and z_i. We use the following three levels of Type A error:
$p_1 = 0.00$

$p_2 = 0.10$
$p_3 = 0.20$

After this error is added the points will generally go off (inside or outside) the surface of the sphere in an uncontrolled manner. We believe it is best to separate the effect (a) of the points being irregularly spaced from (b) their being on or off the surface of the sphere. We therefore make sure that every point is still on the sphere's surface by applying the following equations:

$$x_i'' = x_i' \Big/ \sqrt{x_i'^2 + y_i'^2 + z_i'^2}$$

$$y_i'' = x_i' \Big/ \sqrt{x_i'^2 + y_i'^2 + z_i'^2}$$

$$z_i'' = z_i' \Big/ \sqrt{x_i'^2 + y_i'^2 + z_i'^2}$$

Then we add to the resulting coordinates (all defining points on the surface of the sphere) what we have designated as Type B error. Type B error lets the points go inside or outside the sphere in amounts that we can control. The movement from the surface of the sphere would otherwise be totally uncontrolled had the previous step not been taken. We have designated 5 levels of Type B error: ± l_j where $l_j = 0.20, 0.10, 0.05, 0.01$, and 0.00 for $j = 1 \dots 5$ (the last error level, l_5 , of course corresponds to points being confined to the surface of the sphere; combined with level 1 of Type A error we have the errorless regularly spaced sphere). The error is added as in the following formulae:

$$x_i''' = ((1 - l_j) + 2 \cdot l_j \cdot \beta(x_i))x_i''$$

$$y_i''' = ((1 - l_j) + 2 \cdot l_j \cdot \beta(y_i))y_i''$$

$$z_i''' = ((1 - l_j) + 2 \cdot l_j \cdot \beta(z_i))z_i''$$

Here $\beta\,(v_i) \sim$ i.i.d U[0,1], where U[0,1] is the uniform distribution in the interval [0,1]. The variable l_j is the Type B error level ranging from 0.00 to 0.20.

Evaluation of Mapping Performance

Although PARAMAP reports the κ value at the end of the analysis, and lower values mean better results with an absolute minimum known to be at 1.0, since there is no distribution theory for κ values (conditioned on input data, dimensionality of solution, etc.) we still needed another metric to be able to compare different solutions or even different computer programs (such as ISOMAP) which are based on entirely different principles. For this purpose we developed a metric based on the *k-NN* (*k*-nearest neighbors) approach. De Backer et al. (1998) used a slightly different measure in their comparisons between mapping algorithms and named their metric "classification rate". We will call our metric "rate of agreement in local structure" abbreviated as "agreement rate" or "*A*". The index we developed is pretty similar to the RAND index (Rand, 1971; Hubert & Arabie, 1985). However the RAND index is used to compare two methods of partitioning, based on the number of objects placed in common classes of the respective partitions. *K*-nearest neighbors can also be regarded as a clustering scheme, but there will be many overlapping clusters and thus the resulting clusters will not be partitions. Because of this we cannot readily use the RAND index. The calculation of the agreement rate goes as follows:

1. For every point i find the k-nearest neighbors, both for the input matrix **Y** and the output matrix **X**. There will be two $k \times n$ matrices (neighborhood matrices) that need to be computed for this purpose. The columns will contain the indices of the k closest neighboring points corresponding to column number j. More formally, if we call this matrix **P**, p_{ij} will contain the index of the i^{th} closest point to point j.

2. Let a_i stand for the number of points that are in the k-nearest neighbor list for point i in both **X** and **Y**.

3. The agreement rate will be equal to $\sum_{i=1}^{n} a_i \Big/ kn = A$.

Also after giving this rate we have programmed the routine to randomize the points in the output configuration a given number of times (we initially set this number at 1000) and calculate the same index for each of the randomized configurations with the input configuration fixed. We compute the mean, maximum, and minimum of the agreement rates resulting from randomized configurations of the solution, while also tabulating the empirical distribution of this value. This will provide an estimate of the empirical distribution of the agreement rate for a given **Y** and randomized configurations, constraining the distribution of points, and therefore of distances, in those configurations to be the same as those for **X**.

Another option this routine accomplishes is to produce a plot of δ_{ij} vs. d_{ij} but only for those i and j that are contained in the neighborhood matrix of the output space. This plot is analogous to a Shepard diagram (without monotone regression being used to estimate pseudo-distances) but only for “local” pairs of points as defined via the output configuration. This “Shepard-like” diagram, as we call it, will provide a graphical indication of the smoothness of the relationship between the input distances and the distances derived from the output configuration by showing how close this mapping is to being locally linear or locally monotone. The Shepard-like diagrams for two cases of Paramap mappings of the sphere (the “best” case and the “worst case” in the sense to be defined) can be seen in Figure 9. We also show such a diagram for what we regard as the best mapping configuration of the regularly spaced errorless sphere by ISOMAP.

Problem of Similarity Transformations

One aspect of this overall approach that should be mentioned is that, PARAMAP (and ISOMAP or ISOMAP-like approaches), like other MDS approaches based on the Euclidean metric, determines solutions only up to a similarity transform. In particular, even though the translation of origin and overall scale factor problem can easily be solved for via normalizing conventions, there remains a very severe and important problem of orthogonal rotation of the obtained structures. This problem often makes it very difficult to determine whether the solutions actually obtained are comparable to those one expects to obtain, especially if the representation is in two or more dimensions. We have solved this problem by using some standard approaches for orthogonal rotation of such configurations to optimal congruence with one landmark solution that we obtain, such as the configuration of the two-dimensional solution to the errorless and regularly spaced points on the sphere as given in Figure 7 (Rohlf & Slice 1989; Slice 1994)[4]. After such rotation, we find the solutions obtained using the irregularly spaced "noisy" (or errorful) data are remarkably similar to those for the regularly spaced and error-free configurations with which we began. We also report for each of the rotated configurations a VAF (variance accounted for) measure which is calculated as follows:

$$VAF = 1 - \frac{\sum_{r=1}^{R}\sum_{i=1}^{n}(\hat{x}_{ir} - x_{ir})^2}{\sum_{r=1}^{R}\sum_{i=1}^{n}(x_{ir})^2},$$

where $\hat{x}_{ir}$ belong the configuration obtained via a similarity transformation of any of the solutions for the varying error levels and irregularity of spacing and x_{ir} belong to the landmark configuration, i.e., the solution to the errorless and regularly spaced sphere. There is a caveat, however, about these VAF values. Because of the nature of the PARAMAP algorithm, two different solutions can look exactly the same, though the placement of points will be arbitrary on either of the five parallels, depending on where the sphere has been cut. Thus, although visual inspection reveals the solutions are very much like one another, the VAF values of one might turn out to be lower. The effect of the similarity transformation can be most readily seen by observing the south and north poles (namely points 1 and 62) and observing that the lower numbered points are around point 1 and the higher numbered points are near point 62. To correct for the problem of arbitrary placement of points on the five parallels, we need to supplement the optimal similarity transformation found by Rohlf and Slice (1989) algorithm with another step in which the points on each parallel of the transformed configuration are reordered (but constrained to be a cyclical permutation of the order corresponding to the order for each parallel in the target configuration of the regularly spaced errorless sphere), and alternate this step and the transformation step until convergence occurs. This alternating procedure has not yet been implemented, but we plan to do so in the very near future.

Results

The results were generally very good. The plot of the output configuration of the analysis done on the perfect sphere is given in Figure 7. As can be seen after comparing this solution to Figure 2b, this solution is very reminiscent of the azimuthal equidistant projection of the earth. This particular solution has a κ value of 1.137 and has an agreement rate of 82.9% with $k = 5$. For this particular k, randomizations of **X** have shown that the average agreement rate is only 8.1% while the minimum and maximum rates are respectively 3.5% and 16.7%. The standard deviation of the agreement rates obtained from the randomization procedure is respectively 1.9%. We can use Chebychev's inequality to determine an upper bound on the probability of obtaining a rate of 82.9% simply by chance. If we consider Z to be the random variable representing the distribution of the agreement rate, and σ to be the estimate of the standard

[4] The program we use is called GRF-ND. We use the option of ordinary least-squares (or also called ordinary procrustes) superimposition. This method fits several specimens to a given reference configuration. The superimposition parameters are chosen to minimize the sum of squared distances between points on each configuration and the corresponding points on the reference specimen. The documentation (Slice, 1994) and the program are available online from http://life.bio.sunysb.edu/morph/soft-super.html.

deviation of the distribution, Chebychev's inequality puts an upper bound on the probability that Z can be above the mean μ by some multiple k of σ as in the below equation:

$$P(|Z-\mu| \geq k\sigma) \leq \frac{1}{k^2}$$

For our case 82.9% is above the mean by about 40 standard deviations (39.88 to be exact without rounding in the calculations). Using the above formula, the upper bound probability can be calculated as roughly $1/40^2$ or 0.000625. So the rate of 82.9% is very difficult to achieve by chance alone and can be stated as being very statistically significantly different from that for a random configuration with the same distribution of distances (d_{ij}'s) at a p value << .001. The results are similar also for the other error levels and differing irregularities of spacing.

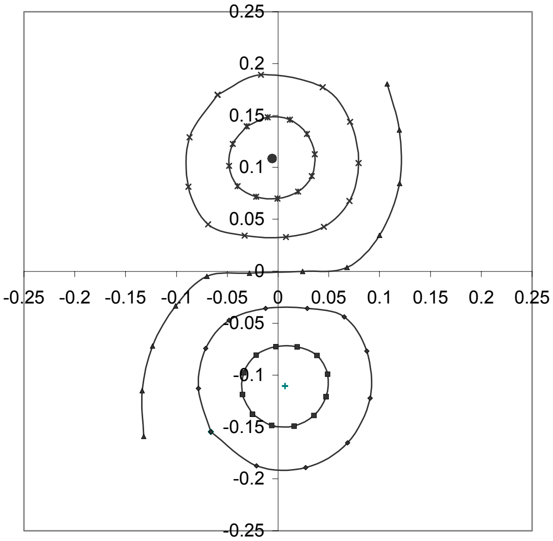

Figure 7
The plot of the solution obtained by running PARAMAP on the 62 points on the sphere. The *kappa* value for this solution is 1.137. Contours represent the 5 equally spaced parallels.

The κ values and the agreement rates of the 15 different error combinations are given in Tables 1 and 2. Tables 3, 4, and 5 provide the minimum, maximum and the average values of agreement rates for the 1000 randomizations of each solution. It is clear from these tables that the obtained agreement rates are much larger than that which could be obtained by chance alone. The plots of all the PARAMAP recovered configurations for the 15 cases are also given in Figures 8a through 8o. For each plot also a measure of congruence (VAF, as explained in the preceding section) of the configuration with the reference configuration (configuration 1 or the two-dimensional solution for the errorless regularly spaced sphere) is provided (but, we remind the reader without the required reordering of the points on the different parallels of the transformed solution). To observe the smoothness of the relationship between the original distances and those obtained by PARAMAP, we present the modified Shepard diagrams given in Figure 9a and 9b for the two extreme cases, respectively, the errorless and regularly spaced sphere and the sphere with the maximum amount of error and the maximum amount of irregularity. Also in Figure 9c we provide the equivalent diagram for the ISOMAP solution to the errorless and regularly spaced sphere, which clearly shows a random distribution of points, unlike the previous two plots. The PARAMAP program was run from 100 different random starts for 200 iterations and then the best solution was further processed for 1000 more iterations for each of these 15 combinations. Although better solutions for each are probable, they would very likely not differ materially from the results we obtained.

Table 1
The κ values for the 15 experimental configurations of the sphere with 62 points

Type B Error Levels (±)	Type A Error Levels		
	0%	10%	20%
0.00	1.137	1.103	1.059
0.01	1.144	1.096	1.046
0.05	1.139	1.109	1.073
0.10	1.128	1.096	1.080
0.20	1.152	1.130	1.121

Table 2
The agreement rate for the 15 experimental configurations of the sphere with 62 points, all given as percentages (*k* = 5)

Type B Error Levels (±)	Type A Error Levels		
	0%	10%	20%
0.00	82.9	79.7	83.5
0.01	78.4	83.2	83.2
0.05	80.0	79.0	88.1
0.10	83.2	82.9	83.0
0.20	80.6	81.0	83.9

Table 3
The minimum of the agreement rate for the 15 experimental configurations of the sphere with 62 points obtained over 1000 randomized configurations of the output matrix, all given as percentages (k = 5)

Type B Error Levels (±)	Type A Error Levels		
	0%	10%	20%
0.00	3.5	2.6	2.9
0.01	2.3	2.6	3.2
0.05	2.9	2.6	3.2
0.10	2.9	2.6	2.9
0.20	2.6	3.5	2.9

Table 4

The average of the agreement rate for the 15 experimental configurations of the sphere with 62 points obtained over 1000 randomized configurations of the output matrix, all given as percentages (k = 5)

	Type A Error Levels		
Type B Error Levels (±)	0%	10%	20%
0.00	8.1	8.1	8.1
0.01	8.2	8.2	8.1
0.05	8.2	8.1	8.2
0.10	8.2	8.1	8.1
0.20	8.1	8.2	8.1

Table 5

The maximum of the agreement rate for the 15 experimental configurations of the sphere with 62 points obtained over 1000 randomized configurations of the output matrix, all given as percentages (k = 5)

	Type A Error Levels		
Type B Error Levels (±)	0%	10%	20%
0.00	16.7	14.5	15.2
0.01	14.2	16.5	15.5
0.05	13.9	15.5	14.2
0.10	14.2	14.2	15.2
0.20	14.5	15.8	15.5

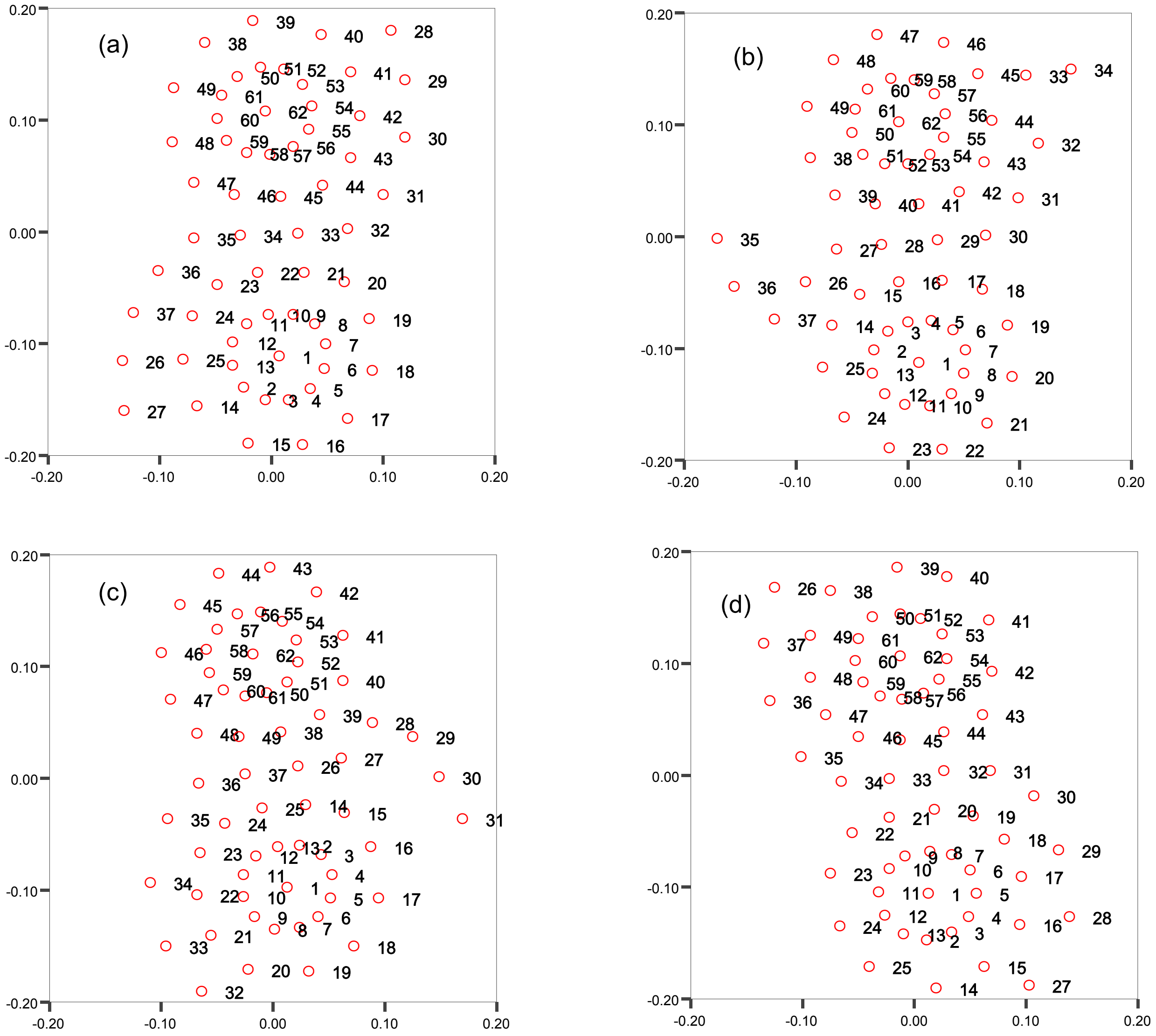

Figure 8
(a) 0% type A error / 0.00 type B error, (b) 0% type A error / 0.01 type B error VAF=0.47, (c) 0% type A error / 0.05 type B error VAF=0.42, (d) 0% type A error / 0.10 type B error VAF=0.63.

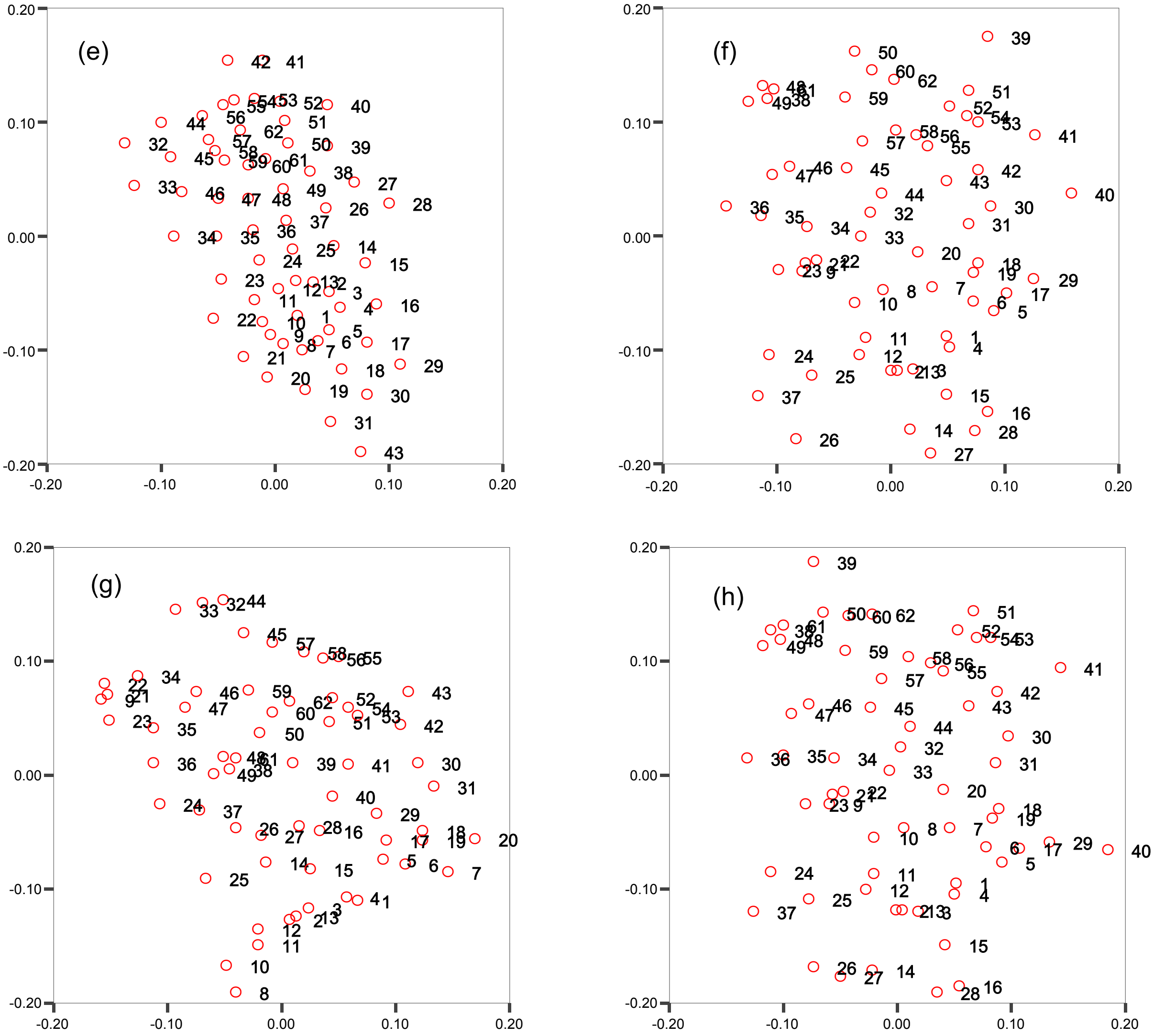

Figure 8 - continued
(e) 0% type A error / 0.20 type B error VAF=0.23, (f) 10% type A error / 0.00 type B error VAF=0.61, (g) 10% type A error / 0.01 type B error VAF=0.33, (h) 10% type A error / 0.05 type B error VAF=0.62.

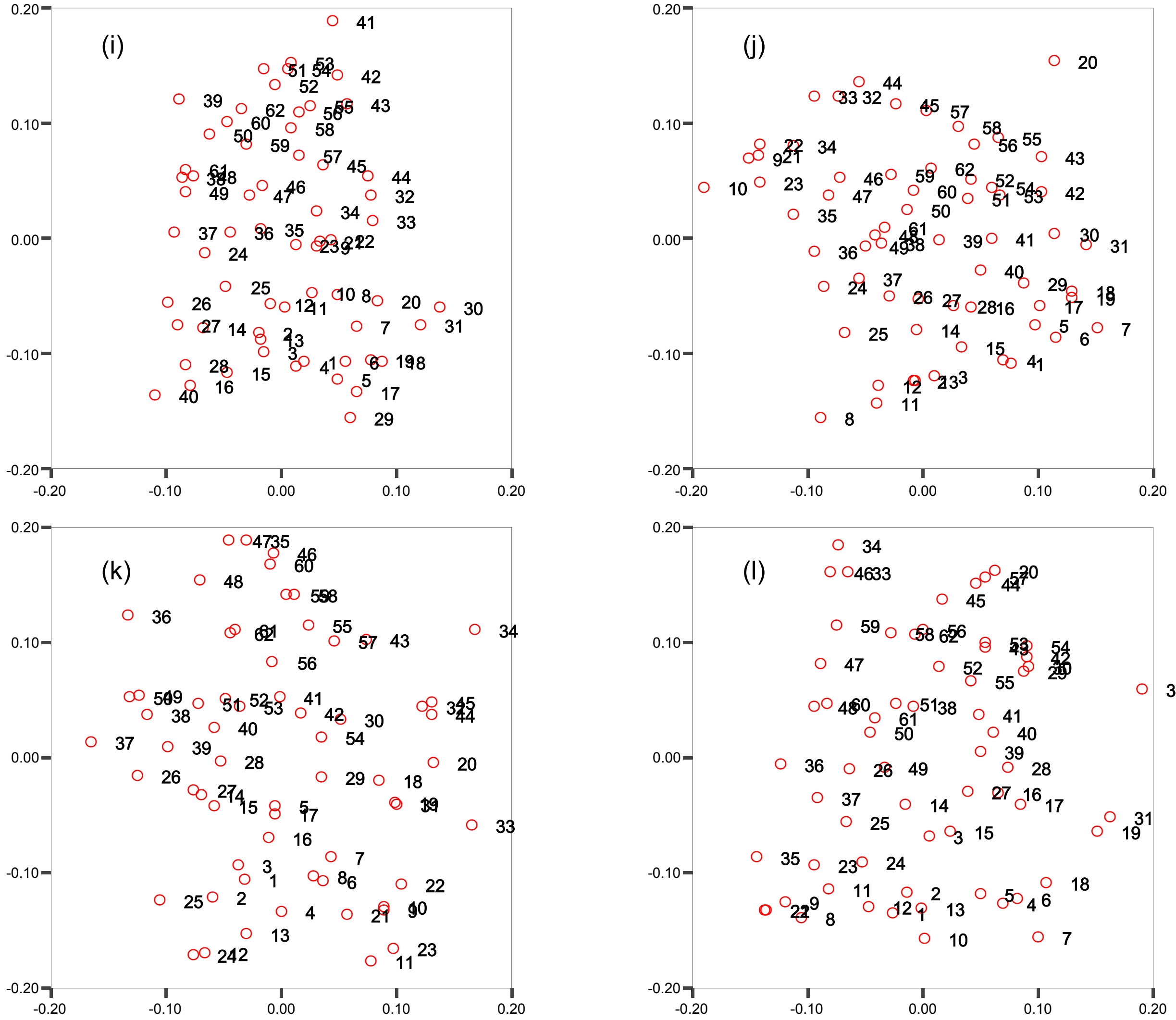

Figure 8 - continued
(i) 10% type A error / 0.10 type B error VAF=0.49, (j) 10% type A error / 0.20 type B error VAF=0.27, (k) 20% type A error / 0.00 type B error VAF=0.37, (l) 20% type A error / 0.01 type B error VAF=0.42.

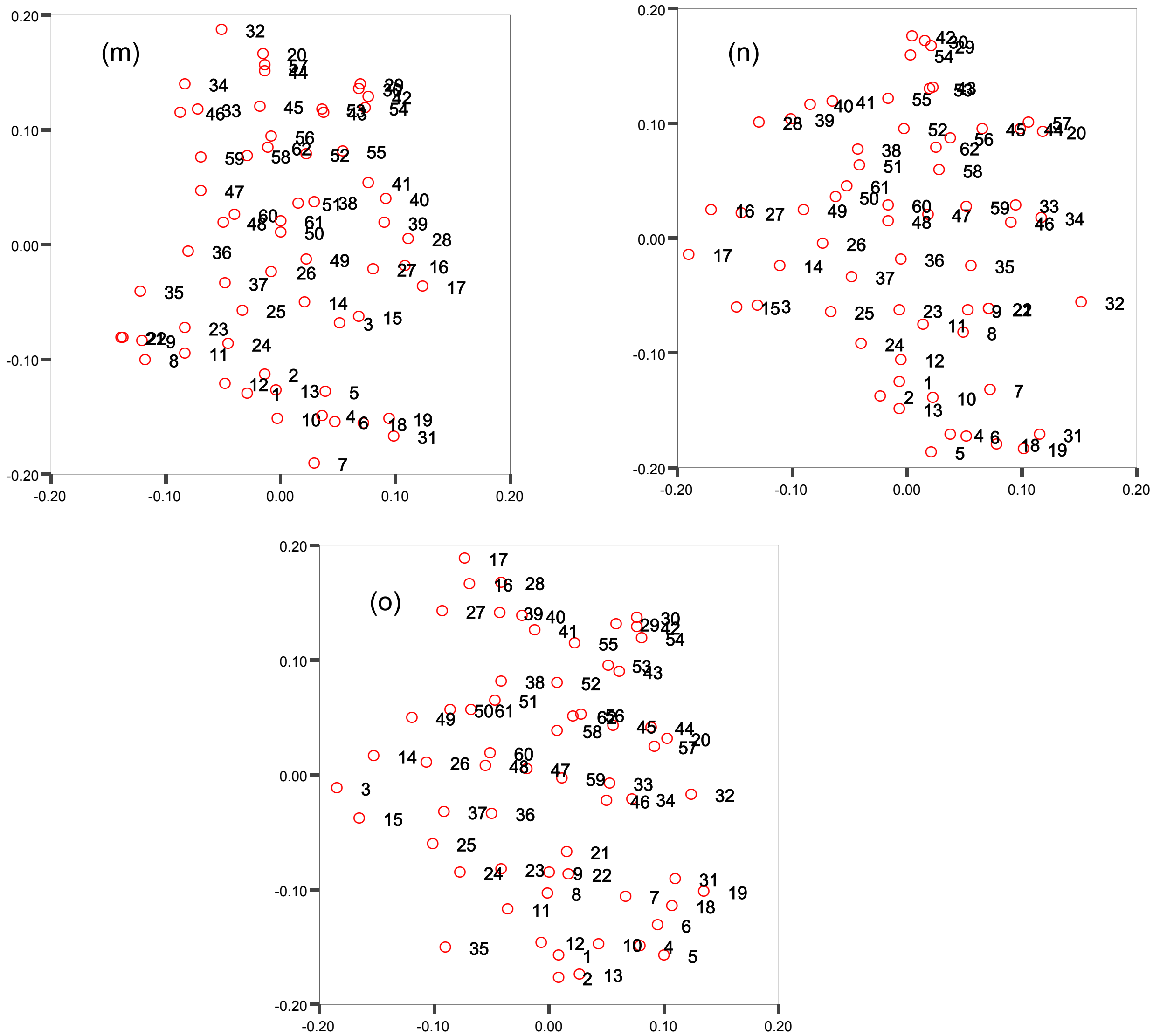

Figure 8 - continued
(m) 20% type A error / 0.05 type B error VAF=0.34, (n) 20% type A error / 0.10 type B error VAF=0.37, (o) 20% type A error / 0.20 type B error VAF=0.36.

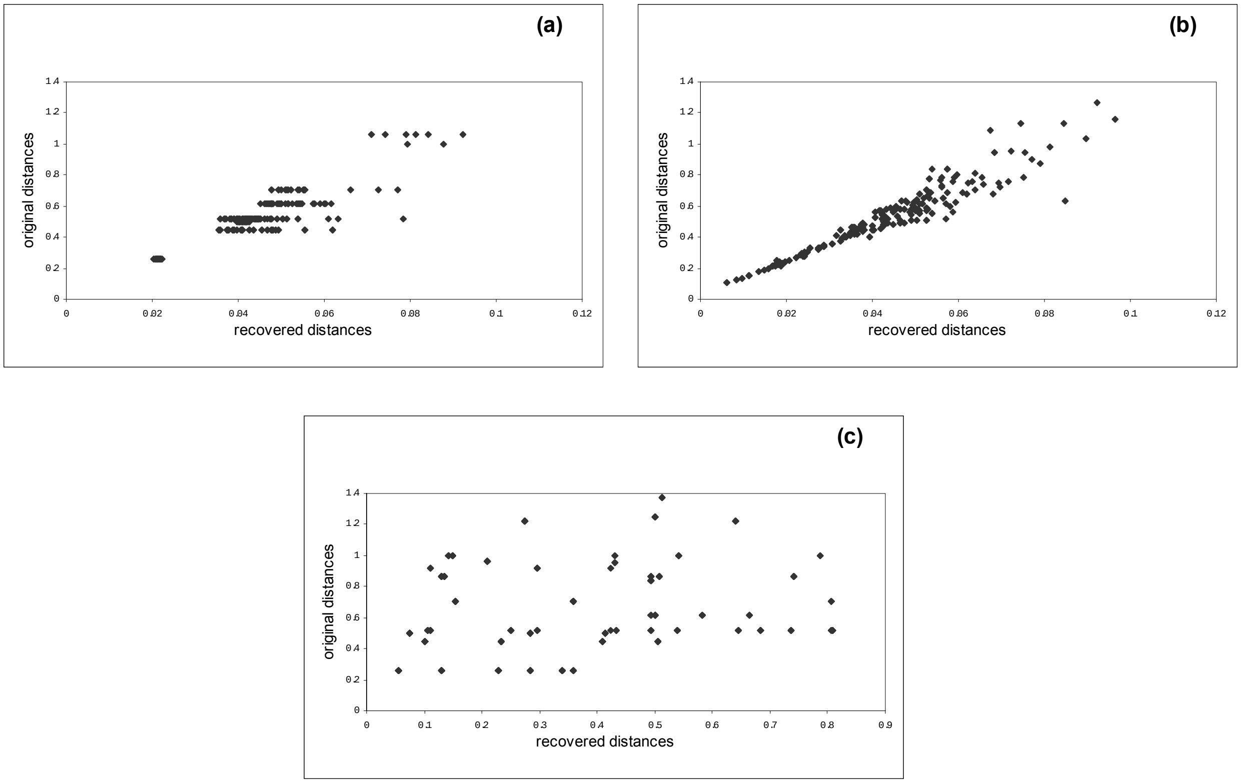

Figure 9
Shepard-like diagrams of PARAMAP solutions to (a) the errorless and regularly spaced sphere (i.e. minimum error), (b) the sphere with 20% type A and ±0.20 Type B error (i.e. maximum error), (c)The Shepard-like diagram of the ISOMAP solution to the sphere with 62 regularly spaced and error free points In all three cases the pairs of indices involved are those *i* and *j* that are contained in the *k* nearest neighborhood matrix of the output space configuration (*k*=5).

We also tested ISOMAP on the regular sphere with no error using the "ε" option and specifying the value of "ε" as 0.550. The reason we used 0.550 is that it is the smallest value making sure that the resulting graph will be connected. Using "ε" as 2.0, the maximum distance in the data, will of course make it equivalent to metric MDS. It is possible to vary the value between the bounds. The results from ISOMAP for the sphere with 62 points are not satisfactory compared to the PARAMAP output. First and foremost, they are merely a projection of the sphere onto the two-dimensional plane. Apart from visually resembling looking through a transparent sphere with the points located as dots on the surface (which is contrary to what the program aims to do, namely identify the true geodesic paths between the points), the agreement rates are also lower (the ISOMAP produced plot is given in Figure 10 with the neighborhood graph drawn in). The agreement rate for the solution shown in Figure 10 is 48.1%. This number makes intuitive sense, since the projection will enforce half the points in the proximity of a point coming from the opposite side of the sphere. It should also be noted that the working of ISOMAP relies on a wise selection of both the *k* and the ε criteria. The worst case, if the values are not carefully selected, is that the graph will be disconnected and ISOMAP will only produce an embedding for those points it can connect. Another problem is that very degenerate solutions can also be obtained for some values, such as a line or several intersecting lines.

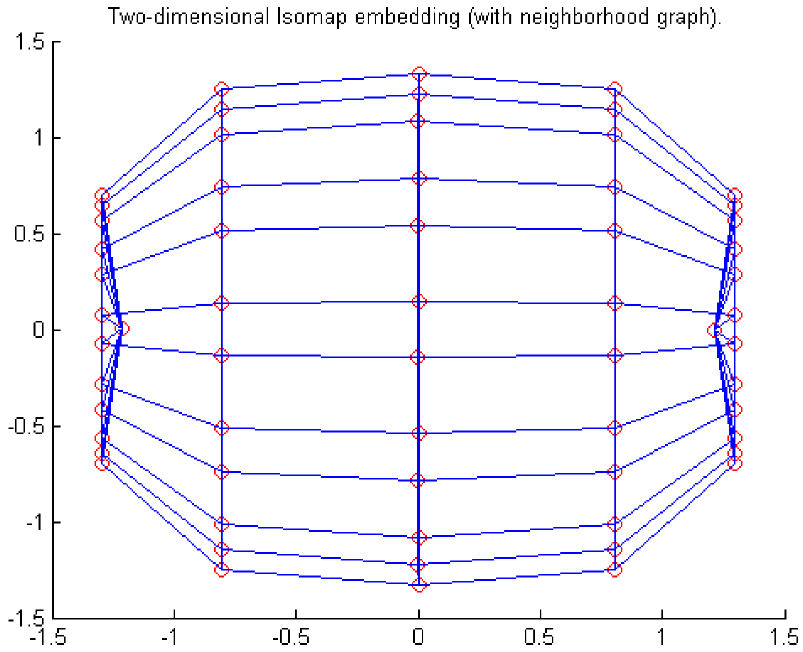

Figure 10
The ISOMAP solution of mapping the 62 points on the sphere with ε = 0.550.

For the Swiss roll data set, on which Tenenbaum et al. (2000) based most of their research on, we also tried a comparison of both approaches. In this case, however, we took a subset of 130 points from the Swiss roll data set that contains 20,000 points. The PARAMAP and ISOMAP mappings are shown in Figures 11(a) and (b), respectively. The agreement rate for the PARAMAP solution is 70.5%, while that rate for the ISOMAP solution is 59.7%. The agreement rate for ISOMAP has improved over the same results with the sphere, suggesting that something more than a projection has taken place; however the PARAMAP result is still better. The map also reveals a property of ISOMAP that tends to be a problem when a smaller number of points is used, namely that the mapping leaves large unfilled gaps while PARAMAP mapping produces a much more homogeneous plot.

In run time it takes about 5 minutes for PARAMAP to complete 100 random starts each with 200 iterations for the 62 points on the sphere. This number reaches 20 minutes for the 130 points on the Swiss roll, suggesting naturally that the computation time scales with the number of objects squared, namely n^2. It must be noted, though, that dimensionality can also play a role, so the actual scale may very likely involve a factor involving R. The run time of ISOMAP in both cases is less than a minute.

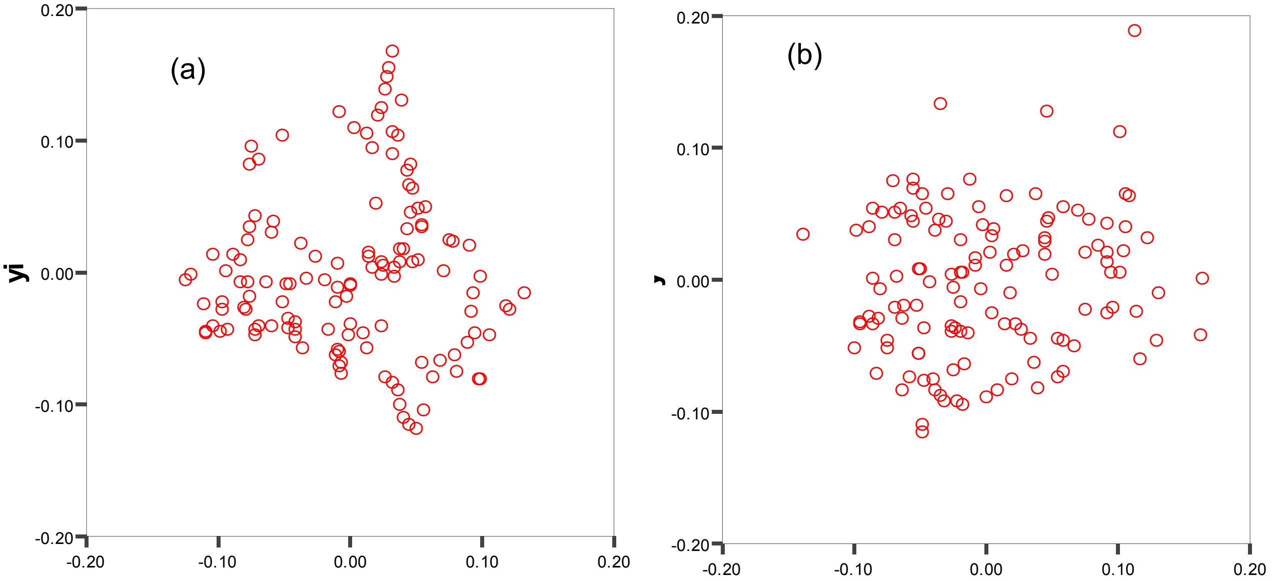

Figure 11
The (a)ISOMAP and (b)PARAMAP solutions to the 130 points on the Swiss roll.

Discussion and Future Direction

One critical feature of PARAMAP that needs to be worked on is its inability to converge to a globally optimal solution in reasonable time. One compromise is to settle for a smaller number of random starts and use a substandard local minimum solution. However, we believe the convergence rate can be enhanced by modifying the steepest descent method used. Indeed, we tried a version of the conjugate gradient method (Fletcher & Reeves, 1964) with the quadratic line search technique (Bertsekas, 1999; Fletcher, 1987). Unfortunately, the cost of the line search was much too expensive and the program was running even slower than the steepest descent version. One possible remedy to the cost of line search is using a combination of both approaches, such as the conjugate gradient method with restarts (Powell, 1977). In this approach a steepest descent iteration is performed every *n* iterations or whenever a certain criterion is met. Another approach might be performing steepest descent iterations until a point is reached where the function can be better approximated by a quadratic function, then reverting to the conjugate gradient method. In any case finding the right optimization procedure requires extensive experimentation as the parameters to be used in each method need to be calibrated. If faster convergence is possible, the global optimum can be attained much more quickly and reliably, with more random starts being permitted for the same computational time.

Another feature to be improved and that is also a drawback of ISOMAP is the lack of a technique that judiciously selects landmarks (rather than randomly selecting them) and after performing the analysis interpolates or extrapolates the remaining points. Decreasing the number of points which will be used as input to the optimization algorithm will result in allowing very large data sets to be analyzed by either of the algorithms.

We believe that after some modifications the PARAMAP algorithm will be a very useful tool in dimensionality reduction, especially when the nonlinearity of the data set in question is very high. ISOMAP can also be useful, but in the limited context of certain kinds of data sets that are not “closed” or nearly so. Combining features of both approaches could result in even more effective and efficient algorithms for nonlinear mapping. Both methods could be modified along lines explained above and better procedures can be devised as a result of this research.

References

Baldi, B., & Hornik, K. (1989). Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks,* 2, 53-58.

Bertsekas, D. P. (1999). *Nonlinear Programming*. Belmont, MA: Athena Scientific.

Buja, A., & Swayne, D. F. (2002). Visualization methodology for multidimensional scaling. *Journal of Classification*, 19, 7-43.

Carroll, J. D. (1968). *Parametric mapping of similarity data as a scaling method*. Unpublished Manuscript, Bell Laboratories, Murray Hill, NJ.

Carroll, J.D., & Chang, J. J. (1968). *PARAMAP, a computer program which performs Parametric Mapping*. Unpublished manuscript, Bell Laboratories, Murray Hill, NJ.

De Backer, S., Naud, A., & Scheunders P. (1998). Non-linear dimensionality reduction techniques for unsupervised feature extraction. *Pattern Recognition Letters,* 19, 711-720.

Deetz, C. H., &Adams, O. S. (1945). *Elements of map projection*. U.S. Department of Commerce Special Publication No. 68, U.S. Government Printing Office, Washington, DC.

Dijkstra, E. W. (1959). A note on two problems in connection with graphs. *Numerische Mathematik,* 1, 269-271.

Fletcher, R. (1987). *Practical Methods of Optimization*. Chichester, U.K.: Wiley.

Fletcher, R., & Reeves, C. M. (1964). Function minimization by conjugate gradients. *Computer Journal*, 7, 149-154.

Floyd, R. W. (1962). Algorithm 97: Shortest Path. C*ommunications of the ACM,* 5, 345.

Foster, I. (1995). *Designing and Building Parallel Programs: Concepts and Tools for Parallel Software Engineering*. Reading: MA: Addison-Wesley [Also available online at http://www-unix.mcs.anl.gov/dbpp].

Hubert, L., & Arabie, P. (1985). Comparing Partitions. *Journal of Classification,* 2, 193-218.

Hu, T. C. (1967). Revised matrix algorithms for shortest paths. *SIAM Journal on Applied Mathematics,* 15, 207-218.

Hu, T. C. (1968). A decomposition algorithm for shortest paths in a network. *Operations Research,* 16, 91-102.

Hu, T. C., & Torres, W. T. (1969). Shortcut in the decomposition algorithm for shortest paths in a network. *IBM Journal of Research and Development*, 13, 387-390.

Kramer, M. A. (1991). Nonlinear principal components analysis using auto-associative neural networks. *AIChE Journal*, 37, 233-243.

Lee, J. A., Landasse, A., & Verleysen, M. (2002). Curvilinear distance analysis versus ISOMAP. In M. Verleysen (ed.), *ESANN’2002 proceedings – European Symposium on Artificial Neural Networks (pp. 185-192).* Bruges, Belgium: D-Side Publications.

Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE, 78,* 1464-1480.

Kohonen, T. (1995). *Self-Organizing Maps*. Berlin: Springer-Verlag.

Kumar, V., Grama, A., Gupta, A., & Karypis, G. (1994). *Introduction to Parallel Computing: Design and Analysis of Algorithms*. Redwood City, CA: Benjamin/Cummings.

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

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

Kruskal, J. B. (1971). Comments on “a nonlinear mapping for data structure analysis”. *IEEE Transactions on Computers,* 20, 1614.

Kruskal, J. B. (1975). Locally Linear and Locally Isometric Mapping. In, *Theory, Methods and Applications of Multidimensional Scaling and Related Techniques, Proceedings of a meeting sponsored by National Science Foundation and the Japan Society for the Promotion of Science*, (pp. 27-33).

Kruskal, J. B., & Carroll, J. D. (1969). Geometric Models and Badness of Fit Functions. In P. R. Krishnaiah (Ed.), *Multivariate Analysis* (pp. 639-671). New York, NY: Academic Press.

Kruskal, J. B., Young, F. W., & Seery, J. B. (1977). *How to use KYST2, a very flexible program to do multidimensional scaling and unfolding*. Murray Hill, NJ: AT & T Bell Laboratories.

Mardia, K. V., Kent, J. T., & Bibby, J. M. (1979). *Multivariate Analysis*. London, England: Academic Press.

Powell, M. J. D. (1977). Restart procedures for the conjugate gradient method. *Mathematical Programming*, 12, 241-254.

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association,* 66, 846-850.

Rohlf, F. J., & Slice, D. E. (1989). *GRF: generalized rotational fitting.* Department of Ecology and Evolution, State University of New York at Stony Brook.

Roweis, S. T., & Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science,* 290, 2323-2326.

Sammon, J. W. (1969). A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers,* 20,

Shepard, R. N., & Carroll, J. D. (1966). Parametric representation of nonlinear data structures. In P. R. Krishnaiah (Ed.), *Multivariate Analysis* (pp. 561-592). New York, NY: Academic Press.

Slice, D. E. (1994). *GRF-ND Generalized rotational fitting of n-dimensional landmark data.* Department of Ecology and Evolution, State University of New York at Stony Brook.

Tenenbaum, J. B., DeSilva, V., & Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science,* 290, 2319-2323.

Torgerson, W. S. (1958). *Theory and Methods of Scaling*. New York, NY: Wiley.

von Neuman, J. (1941). Distribution of the ratio of the mean square successive difference to the variance. *The Annals of Mathematical Statistics,* 12, 367-395.

von Neuman, J., Kent, R. H., Bellinson, H. R., & Hart, B. I. (1941). The mean square successive difference. *The Annals of Mathematical Statistics,* 12, 153-162.

Webb, A. R. (1995). Multidimensional scaling by iterative majorization using radial basis functions. *Pattern Recognition,* 28, 753-759.