Tree preserving embedding

Albert D. Shieh, Tatsunori B. Hashimoto, and Edoardo M. Airoldi[1]

Department of Statistics, Harvard University, Cambridge, MA 02138

Edited* by Stephen E. Fienberg, Carnegie Mellon University, Pittsburgh, PA, and approved August 19, 2011 (received for review December 8, 2010)

The goal of dimensionality reduction is to embed high-dimensional data in a low-dimensional space while preserving structure in the data relevant to exploratory data analysis such as clusters. However, existing dimensionality reduction methods often either fail to separate clusters due to the crowding problem or can only separate clusters at a single resolution. We develop a new approach to dimensionality reduction: tree preserving embedding. Our approach uses the topological notion of connectedness to separate clusters at all resolutions. We provide a formal guarantee of cluster separation for our approach that holds for finite samples. Our approach requires no parameters and can handle general types of data, making it easy to use in practice and suggesting new strategies for robust data visualization.

hierarchical clustering | multidimensional scaling

Visualization is an important first step in the analysis of high-dimensional data (1). High-dimensional data often has low intrinsic dimensionality, making it possible to embed the data in a low-dimensional space while preserving much of its structure (2). However, it is rarely possible to preserve all types of structure in the embedding. Therefore, dimensionality reduction methods can only aim to preserve particular types of structure. Linear methods such as principal component an alysis (PCA) (3) and classical multidimensional scaling (MDS) (4–6) preserve global distances, while nonlinear methods such as manifold learning methods (7–9) preserve local distances defined by kernels or neighborhood graphs. However, most dimensionality reduction methods fail to preserve clusters (10), which are often of greatest interest.

Clusters are difficult to preserve in embeddings due to the so-called crowding problem (11). When the intrinsic dimensionality of the data exceeds the embedding dimensionality, there is not enough space in the embedding to allow clusters to separate. Therefore, clusters are forced to collapse on top of each other in the embedding. As the embedding dimensionality increases, there is more space in the embedding for clusters to separate and the crowding problem disappears, making it possible to preserve clusters exactly (12). However, because the embedding dimensionality is at most two or three for visualization purposes, the crowding problem is prevalent in practice. When the clusters are known, they can be used to guide the embedding to avoid the crowding problem (13). However, the embedding is often used to help find the clusters in the first place. Therefore, it is important to solve the crowding problem without knowledge of the clusters.

Force-based methods such as stochastic neighbor embedding (SNE) (14), variants of SNE (10, 11, 15, 16), and local MDS (17) have been proposed to overcome the crowding problem. Force-based methods use attractive forces to pull together similar points and repulsive forces to push apart dissimilar points. SNE and its variants use forces based on kernels, while local MDS uses forces based on neighborhood graphs. Force-based methods have long been used in graph drawing to separate clusters (18, 19). Although force-based methods are effective, it is difficult to balance the relative strength of attractive and repulsive forces. When repulsive forces are too weak, they will fail to separate clusters, but when repulsive forces are too strong, they will artificially create clusters. Therefore, force-based methods are sensitive to intrinsic resolution parameters such as kernel bandwidths and neighborhood graph sizes that control the amount of separation between points in the embedding.

We introduce tree preserving embedding (TPE) to overcome the limitations of force-based methods. TPE aims to preserve both distances and clusters by preserving the single linkage (SL) dendrogram in the embedding. SL is a hierarchical clustering method that iteratively merges pairs of clusters with minimum nearest neighbor distance. The SL dendrogram is the associated tree with the clusters as vertices and the merge distances as vertex heights. TPE preserves the SL dendrogram in the sense that SL generates the same dendrogram from both the data and the embedding. Embeddings and dendrograms have long been used as complementary representations for dissimilarities (20). However, there is no guarantee that embeddings and dendrograms will be consistent when used separately. In particular, clusters found by dendrograms may not be found in embeddings due to the crowding problem. TPE combines embeddings and dendrograms in a common representation.

Preserving the SL dendrogram in the embedding is a natural choice for several reasons. First, the SL dendrogram is the only dendrogram consistent with the minimum spanning tree (MST) in the sense that the SL dendrograms are the same when the MSTs are the same (21, 22). Preserving the topologies of neighborhood graphs has been shown to help overcome the crowding problem (23). However, while the topologies of neighborhood graphs such as the MST can only be preserved approximately in general (24), we show that the SL dendrogram can be preserved exactly. Second, the SL dendrogram represents both global and local structure due to its hierarchical nature. Preserving global structure allows TPE to separate clusters, while preserving local structure prevents TPE from artificially creating clusters. Finally, TPE can separate clusters even when the SL dendrogram cannot. Although SL is often criticized as a clustering method for finding poor clusters in practice (25, 26), SL finds poor clusters due to the instability of cutting the SL dendrogram at a particular height (27). Because TPE preserves the SL dendrogram at all heights, TPE is not sensitive to the instabilities of the SL dendrogram at any particular height.

We make cluster separation in TPE precise using the topological notion of connectedness (25). A natural and commonly used notion of a cluster is a set of points that are connected at a particular resolution. It is well known that the SL dendrogram finds clusters of connected points at different resolutions for different heights (25). We show that TPE preserves connectedness in the sense that points in the embedding are connected if and only if they are connected in the data. Preserving connectedness guarantees that clusters separated in the data remain separated in the embedding. Therefore, TPE is guaranteed to separate clusters at all resolutions rather than just a single resolution.

Author contributions: A.D.S., T.B.H., and E.M.A. designed research; A.D.S., T.B.H., and E.M.A. performed research; A.D.S., T.B.H., and E.M.A. analyzed data; and A.D.S., T.B.H., and E.M.A. wrote the paper.

The authors declare no conflict of interest.

*This Direct Submission article had a prearranged editor.

[1]To whom correspondence should be addressed. E-mail: airoldi@fas.harvard.edu.

This article contains supporting information online at www.pnas.org/lookup/suppl/doi:10.1073/pnas.1018393108/-/DCSupplemental.

Methods

In this section, we introduce TPE as an optimization problem subject to a set of constraints that preserve the SL dendrogram in the embedding. The constraints arise from a characterization of the SL dendrogram using a notion of connectedness. We introduce an algorithm similar to hierarchical clustering to implement TPE. In order to make the algorithm practical, we propose a variant based on a greedy approximation. Finally, we show that TPE preserves connectedness in a precise sense that corresponds well with separating clusters.

Algorithm. TPE is based on the framework of MDS (28). Given a real, symmetric, nonnegative, zero diagonal $n \times n$ dissimilarity matrix D for a set of n objects $S = \{1,\ldots,n\}$ in a high-dimensional space, MDS finds a p-dimensional Euclidean embedding $X = \{x_1,\ldots,x_n\} \subseteq \mathbb{R}^p$ of the objects that minimizes a loss function such as stress

$$\sigma(X) = \sum_{x_i,x_j \in X} (d_{i,j}(X) - D_{i,j})^2,$$

the sum of squared errors between the Euclidean distances $d_{i,j}(X) = \|x_i - x_j\|$ in the embedding and the corresponding dissimilarities $D_{i,j}$. Because loss functions such as stress emphasize approximating large dissimilarities well, minimizing them without constraints on the embedding leads to the crowding problem. TPE preserves the SL dendrogram in the embedding in order to overcome the crowding problem.

SL is a hierarchical clustering method that iteratively merges pairs of clusters A, $B \subseteq S$ with minimum nearest neighbor distance

$$\Delta(A,B) = \min_{i \in A, j \in B} D_{i,j},$$

starting with the n singleton clusters and ending with the trivial cluster. The SL dendrogram is the associated binary tree of depth $n-1$ with singleton clusters as leaf vertices, the trivial cluster as the root vertex, merged clusters as internal vertices, and merge distances as vertex heights. For an example, see Fig. 1. There are many equivalent characterizations of the SL dendrogram (26). We use the following notion of connectedness to express the SL dendrogram as a set of constraints on pairs of both objects and points.

Definition 1: Objects i, $j \in S$ are ε-connected if there exists a path of objects $\alpha_1 = i,\ldots,\alpha_m = j \in S$ such that $D_{\alpha_l,\alpha_{l+1}} \le \varepsilon$ for $l = 1,\ldots,m-1$.

Definition 2: Points x_i, $x_j \in X$ are ε-connected if there exists a path of points $x_{\alpha_1} = x_i,\ldots,x_{\alpha_m} = x_j \in X$ such that $d_{\alpha_l,\alpha_{l+1}}(X) \le \varepsilon$ for $l = 1,\ldots,m-1$.

Intuitively, objects are connected if there exists a path with short hops between them. The SL dendrogram contains the paths with short hops between objects. Objects are ε-connected if there exists a path of vertices with heights at most ε between their associated leaf vertices, or singleton clusters, in the SL dendrogram. Therefore, cutting the SL dendrogram at a height of ε produces clusters of ε-connected objects (25). The relationship between the SL dendrogram and connectedness in an embedding is illustrated in Fig. 1. The merge distances of clusters in the SL dendrogram determine the connectedness of clusters in the embedding.

Cluster merges connect objects in the SL dendrogram. The ultrametric distance between objects in a dendrogram is the distance at which they are merged into the same cluster (29). The ultrametric distance in the SL dendrogram is equivalent to the maximal subdominant ultrametric distance

$$L_{i,j} = \min_{\alpha_1 = i,\ldots,\alpha_m = j \in S} \max_{l=1}^{m-1} D_{\alpha_l,\alpha_{l+1}},$$

the maximal hop in a minimal path between objects (30), reminiscent of commute times in a graph (31). Because the paths in the MST are minimal paths, the SL dendrogram can be constructed efficiently from the MST in practice (21). However, it is important to emphasize that the paths in the MST are not the only possible minimal paths.

The SL dendrogram can be characterized by two constraints on each pair of objects. First, each pair of objects i, $j \in S$ must be $L_{i,j}$-connected by their ultrametric distance $L_{i,j}$. Second, each pair of objects i, $j \in S$ cannot be ε-connected by any distance ε less than their ultrametric distance $L_{i,j}$. TPE uses the constraints on pairs of objects as constraints on corresponding pairs of points in the embedding. The first constraint guarantees that clusters in the embedding are merged at the same distances as corresponding clusters in the data. The second constraint guarantees that clusters in the embedding are merged in the same order as corresponding clusters in the data.

The following algorithm implements TPE.

1. Initialize the indices of available clusters

$$I_1 = \{1,\ldots,n\},$$

the indices of the singleton clusters

$$S_1 = \{1\},\ldots,S_n = \{n\},$$

the embeddings of the singleton clusters

$$X_1 = \{x_1 = 0\},\ldots,X_n = \{x_n = 0\},$$

and the ultrametric distances for the singleton clusters

$$L_{1,1,1} = 0,\ldots,L_{n,n,n} = 0$$

where $L_{c,i,j} = L_{i,j}$ denotes the ultrametric distance between objects $i, j \in S_c$ contained in cluster c.

2. For each iteration $k = 1,\ldots,n-1$:

i. Find the next cluster merge

$$a_k,b_k = \operatorname{argmin}_{a,b \in I_k:\ a \ne b} \Delta(S_a,S_b)$$

at a merge distance of

$$\Delta_k = \Delta(S_{a_k},S_{b_k}).$$

ii. Merge the clusters

$$S_{n+k} = S_{a_k} \cup S_{b_k}$$

and update the indices of available clusters

$$I_{k+1} = \{i \in I_k:\ i \ne a_k,b_k\} \cup \{n+k\}.$$

iii. Find the ultrametric distances for the merged cluster

$$L_{n+k,i,j} = \begin{cases} L_{a_k,i,j} & \text{if } i,j \in S_{a_k} \\ L_{b_k,i,j} & \text{if } i,j \in S_{b_k} \\ \Delta_k & \text{otherwise} \end{cases} \quad \forall\ i,j \in S_{n+k}.$$

iv. Embed the merged cluster

$$\begin{aligned} &X_{n+k} = \operatorname{argmin}_{X = \{x_i \in \mathbb{R}^p:\ i \in S_{n+k}\}} \sigma(X) \\ &\text{s.t.} \quad x_i,x_j \quad \text{are } L_{n+k,i,j}\text{-connected} \quad \forall\ i,j \in S_{n+k}, \\ &d_{i,j}(X) \ge L_{n+k,i,j} \quad \forall\ i,j \in S_{n+k}. \end{aligned} \qquad [1]$$

3. Return the embedding X_{2n-1}.

The algorithm proceeds similarly to hierarchical clustering. There are $n-1$ iterations, one for each depth of the SL dendrogram. At each iteration, a pair of clusters is merged and the merged cluster is embedded by minimizing stress subject to the connectedness constraints. At the last iteration, the trivial cluster is embedded and returned. The number of objects being

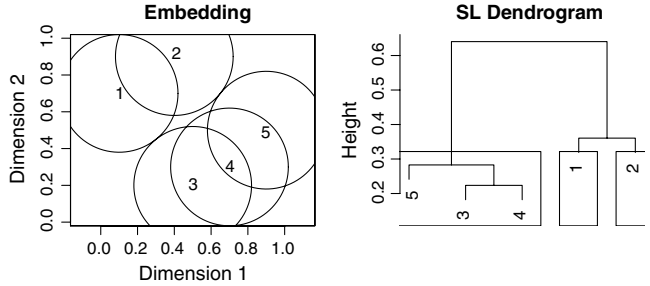

Fig. 1. Relationship between the SL dendrogram and connectedness in an embedding. Cutting the SL dendrogram at a height of $\varepsilon = 0.3$ produces three clusters of ε-connected points. Points 3 and 5 are ε-connected by point 4 because the ε-ball centered at point 4 contains points 3 and 5, while points 1 and 2 are not ε-connected to any other points because they are not contained by any ε-balls centered at other points.

embedded changes at each iteration depending on the size of the merged cluster. Because the embeddings at each iteration are independent, only the embedding at the last iteration is needed. However, earlier embeddings can be used to help initialize later embeddings in practice.

The connectedness constraints can always be fulfilled. In fact, it is trivial to find an embedding that fulfills the connectedness constraints.

Theorem 1. *For each iteration $k = 1,\ldots,n-1$, there is a feasible solution to the optimization problem* ***1****.*

The main difficulty in TPE is minimizing stress. The connectedness constraints may appear to be too rigid to allow TPE to find a low stress embedding because each pair of objects can be connected by an arbitrary path of objects. However, the connectedness constraints do not specify that the paths connecting pairs of points must be the same as the paths connecting corresponding pairs of objects. Preserving the paths would preserve the MST, which is not possible in general (24). Moreover, the flexibility in choosing the paths allows points to move more freely in the embedding to lower stress. However, this flexibility comes at a cost. Due to the combinatorial nature of choosing paths, the optimization problem **1** is computationally intractable. Nevertheless, we can obtain a computationally tractable approximation by restricting the types of paths that can be chosen.

Greedy Approximation. The connectedness constraints allow all points in a merged cluster to be rearranged in the embedding at each iteration. However, because each cluster being merged has already been embedded in prior iterations, it is wasteful to change the prior embeddings of the clusters. The connectedness constraints within the clusters are already fulfilled in their prior embeddings. Because connectedness is preserved under rigid transformations (rotations, translations, and reflections), if we restrict the placement of the clusters to rigid transformations, then we only need to fulfill the connectedness constraints between them. The paths that fulfill the connectedness constraints between the clusters must pass through their nearest neighbors. Therefore, placing the clusters exactly their merge distance apart fulfills the connectedness constraints between them. The remaining flexibility in placing the clusters can be used to minimize the stress between them.

In place of the optimization problem **1**, the greedy approximation proceeds at iteration k as follows.

1. Find a rigid transformation that aligns the clusters while keeping them separated by exactly their merge distance

$$T^* = \operatorname{argmin}_{T\in E(p)} \sum_{x_i\in X_{a_k}, x_j\in X_{b_k}} (d_{i,j}(T) - D_{i,j})^2$$
$$\text{s.t.} \quad \min_{x_i\in X_{a_k}, x_j\in X_{b_k}} d_{ij}(T) = \Delta_k \qquad [2]$$

   where $d_{i,j}(T) = \|T(x_i) - x_j\|$ is the Euclidean distance in the embedding after alignment and $E(p)$ is the set of p-dimensional rigid transformations.
2. Align the clusters

$$x_i = T^*(x_i) \quad \forall\, x_i \in X_{a_k}.$$

3. Return the merged cluster

$$X_{n+k} = X_{a_k} \cup X_{b_k}.$$

The alignment of the clusters in the greedy approximation is illustrated in Fig. 2. The greedy approximation is reminiscent of Procrustes analysis (32), which has been used to merge embeddings of clusters (33). However, Procrustes analysis aligns clusters without constraints on the embedding, making it sensitive to the crowding problem. In contrast, the greedy approximation has a constraint that keeps the clusters separated in the embedding in order to preserve the SL dendrogram. The constraint makes the optimization problem **2** more difficult to solve than standard Procrustes problems. Nevertheless, the optimization problem **2** can be solved efficiently in practice. Details of the greedy approximation can be found in *SI Text*.

The efficiency of the greedy approximation comes at the cost of sensitivity to the merge order of the SL dendrogram. Because the greedy optimization cannot change the shapes of the clusters from prior embeddings, it cannot always align the clusters well. For small cluster merges, the prior embeddings will have little effect on the alignment. However, for large cluster merges, there may not be enough empty space in the prior embeddings to allow the clusters to be aligned well. While there are no formal guarantees for the performance of the greedy approximation, we found that good alignments of the clusters can typically be found in practice.

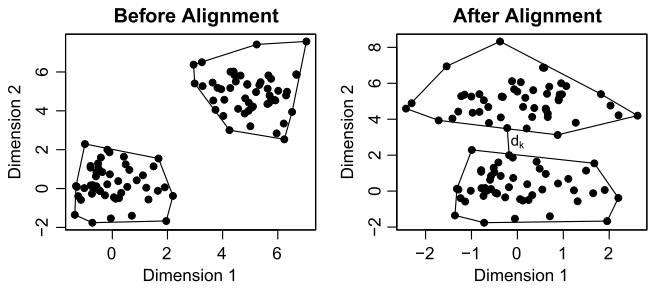

Fig. 2. Alignment of the clusters in the greedy approximation at iteration k. Before alignment, the clusters are placed far apart with high stress. During alignment, one of the clusters is moved using a rigid transformation to minimize the stress between the clusters subject to the constraint that the clusters are placed exactly their merge distance of d_k apart. After alignment, the clusters are placed close together with low stress.

The greedy approximation has a time complexity of $O(n^3)$ because there are $O(n)$ iterations, each of which requires $O(1)$ evaluations of the $O(n^2)$ stress between the clusters in order to find an alignment of the clusters. The greedy approximation has a comparable time complexity to many dimensionality reduction methods, including those based on a spectral decomposition such as classical MDS and many manifold learning methods. While the cubic time complexity of TPE may be prohibitive for some applications, methods developed to improve the scalability of MDS such as landmark points (34) can be extended to TPE in principle.

Connectedness. TPE preserves clusters at all resolutions rather than just a single resolution due to the hierarchical nature of the SL dendrogram. Because the clusters found by cutting the SL dendrogram at different heights can be characterized by connectedness at different resolutions, TPE preserves connectedness in the following sense.

Theorem 2. *For any $\varepsilon > 0$, points x_i, $x_j \in X$ are ε-connected if and only if objects i, $j \in S$ are ε-connected.*

TPE preserves connectedness in the sense that points are connected in the embedding if and only if they are connected in the data. Clusters at any resolution can be neither too close together nor too far apart in the embedding without violating connectedness at some resolution. Therefore, preserving connectedness guarantees that clusters separated in the data remain separated in the embedding. Preserving connectedness is of more than just theoretical interest. Because connectedness applies to finite samples, preserving connectedness provides a formal guarantee of cluster separation for TPE in practice. To our knowledge, TPE is the only method with a formal guarantee of this kind.

Preserving connectedness provides implicit bounds on the Euclidean distances between pairs of points in the embedding. Let x_i, x_j be points merged into the same cluster at iteration k such that $L_{i,j} = \Delta_k$ and define

$$U_{i,j} = \sum_{k' :\ k'\le k, S_{n+k'}\subseteq S_{n+k}} \Delta_{k'},$$

the sum of the merge distances between clusters contained in the merged cluster up to and including iteration k. The Euclidean distance between the points is bounded as follows.

Theorem 3.

$$L_{i,j} \le d_{ij}(X) \le U_{i,j}.$$

Corollary 4.

$$\sigma(X) \le \sum_{x_i, x_j\in X} \max\{(D_{i,j} - L_{i,j})^2, (D_{i,j} - U_{i,j})^2\}.$$

Bounds on the Euclidean distances between pairs of points in the embedding provide a trivial upper bound on the stress of the embedding. Therefore, preserving connectedness prevents TPE from producing embed-

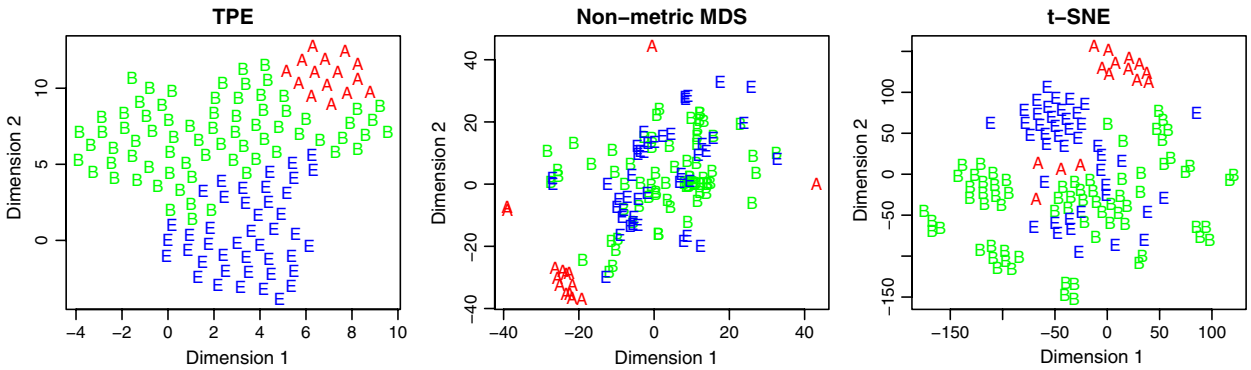

Fig. 3. Embeddings of protein sequences by TPE, nonmetric MDS, and t-SNE. Each point is a protein sequence labeled by the domain it belongs to where A denotes Archaea, B denotes Bacteria, and E denotes Eukaryota.

dings with arbitrarily high stress regardless of the quality of the optimization method such as the greedy approximation.

Results

In this section, we demonstrate the applicability of TPE by analyzing examples drawn from molecular biology, signal processing, and computer vision both qualitatively and quantitatively. Rather than being exhaustive, our goal is to highlight some of the features of TPE through each example. We compare TPE to both classical methods, PCA and nonmetric MDS, and a popular force-based method, t-SNE (11), that recent studies have found separates clusters well (10).

Protein Sequences. In our first example, we analyzed 124 protein sequences of 3-phosphoglycerate kinases (3-PGKs) belonging to the domains Archaea, Bacteria, and Eukaryota collected from public databases by ref. 35. Because protein sequences cannot be represented as real vectors, methods such as PCA that require such a representation cannot be used. Finding a good metric for protein sequences is a difficult and longstanding problem (36). We used inverse sequence alignment scores from the basic local alignment search tool (BLAST) (37) as dissimilarities. Because BLAST scores can be highly nonmetric (12), they are notoriously difficult to embed without collapsing points on top of each other.

We compared TPE to nonmetric MDS, a variant of MDS for nonmetric dissimilarities, and t-SNE in Fig. 3. TPE clearly separates all three domains, while nonmetric MDS and t-SNE mix members of different domains together. Nonmetric MDS collapses many points on top of each other, while TPE spaces the points evenly, reflecting the lack of information in the values of the dissimilarities. t-SNE separates small clusters within each domain but does not preserve their relative locations and mixes them together, while TPE keeps each domain in a contiguous region. Because the merge order of the SL dendrogram is preserved under monotonic transformations of the dissimilarities, TPE is more sensitive to the rank order than the values of the dissimilarities. Therefore, TPE is not as sensitive to nonmetric dissimilarities.

Radar Signals. In our second example, we analyzed 351 radar signals targeting free electrons in the ionosphere collected by ref. 38. Each radar signal consisted of 34 integer and real measurements. We treated each radar signal as a 34-dimensional real vector and used Euclidean distances as dissimilarities. Good radar signals were defined as those that returned evidence of free electrons in the ionosphere, while bad radar signals were defined as those that passed through the ionosphere and returned background noise. Therefore, good radar signals are highly similar, while bad radar signals can be highly dissimilar.

We compared TPE to PCA and t-SNE in Fig. 4. TPE clearly separates good and bad radar signals, while PCA and t-SNE mix them together. PCA collapses the good and bad radar signals on top of each other with little separation. t-SNE separates small clusters of good and bad radar signals but does not preserve their relative locations and mixes them together. TPE keeps good and bad radar signals in contiguous regions. Moreover, TPE concentrates the good radar signals and disperses the bad radar signals, reflecting the different amounts of noise in the radar signals. Therefore, TPE preserves both clusters and density.

Handwritten Digits. In our final example, we analyzed 1,000 images of handwritten digits collected by the United States Postal Service in ref. 39. Each image was 16×16 pixels and grayscale color. We treated each image as a 256-dimensional real vector and used Euclidean distances as dissimilarities. Because the intrinsic dimensionality of handwritten digits is thought to be much higher than two or three (40), it is notoriously difficult to separate all 10 digits in an embedding due to the crowding problem.

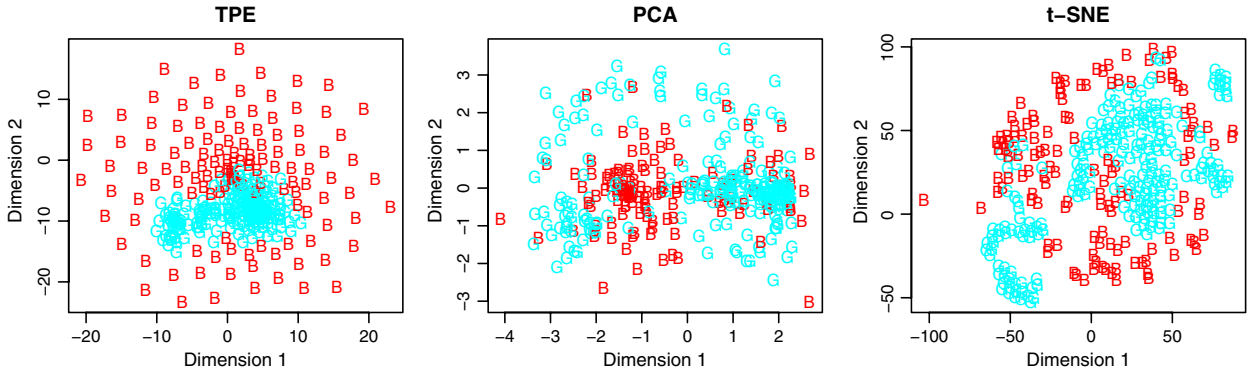

Fig. 4. Embeddings of radar signals by TPE, PCA, and t-SNE. Each point is a radar signal labeled by its quality where G denotes a good radar signal and B denotes a bad radar signal.

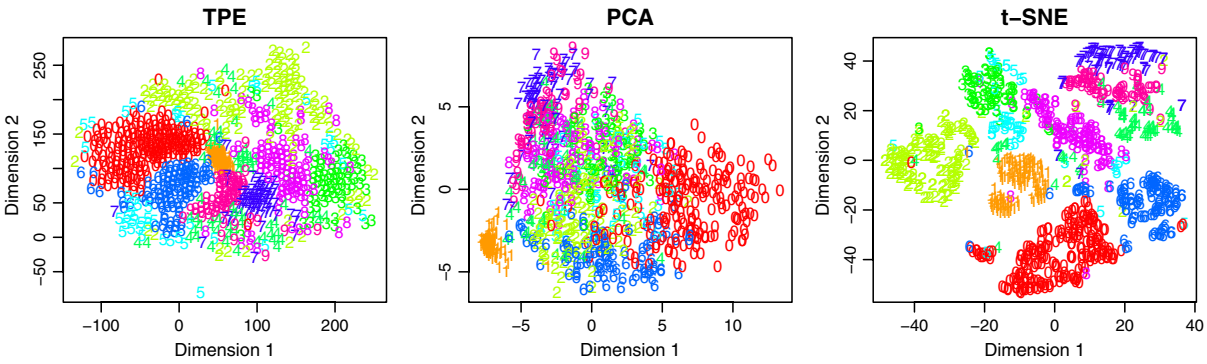

Fig. 5. Embeddings of handwritten digits by TPE, PCA, and t-SNE. Each point is an image labeled by the digit it represents.

We compared TPE to PCA and t-SNE in Fig. 5. TPE and t-SNE separate all 10 digits, while PCA can only separate some of them. t-SNE most clearly separates all 10 digits by creating empty space between them. Because TPE cannot create empty space between clusters without violating connectedness, it is more sensitive than t-SNE to the crowding problem, particularly when there are many clusters. However, t-SNE does not preserve density well. For example, t-SNE separates small clusters within the digit one, while TPE and PCA show that the digit one is the densest cluster, reflecting the minimal amount of variation in how it is written. Although TPE does not separate clusters as well as t-SNE, TPE preserves density better than t-SNE. Therefore, TPE strikes a balance between preserving clusters and density.

Quantitative Evaluation. Because qualitative evaluation of the quality of the embeddings can be subjective, quantitative evaluation is important. We used several popular performance metrics to evaluate the extent to which the embeddings preserve the dissimilarities, the local neighborhoods, and the known clusters. Although TPE sacrifices preserving the dissimilarities by preserving connectedness, the loss is relatively small compared to the gain made in preserving the local neighborhoods and the known clusters, which are widely believed to be more important for visualization purposes (10). Details of the quantitative evaluation can be found in *SI Text*.

Discussion

Revealing clusters is one of the main goals of visualization. However, most dimensionality reduction methods have difficulty preserving clusters due to the crowding problem. In three difficult examples, TPE was able to separate clusters of interest well compared to other dimensionality reduction methods. It is important to emphasize that the success of TPE is not a mere consequence of the ability of the SL dendrogram to separate clusters. In all three examples, cutting the SL dendrogram produced clusters at a single resolution that were no better than random clusters in terms of accuracy with respect to the known clusters. TPE succeeds by preserving clusters at all resolutions rather than just a single resolution.

Dimensionality reduction methods that separate clusters often have issues with artificially creating clusters. It is well known that force-based methods such as t-SNE can find clusters in data where there are none (10). TPE is not as susceptible to this problem. While preserving connectedness prevents clusters from being too close together, it also prevents clusters from being too far apart. Therefore, it is difficult for TPE to artificially create clusters without violating connectedness. In order to empirically test whether TPE artificially creates clusters, we simulated an example of a difficult convex manifold, the Swiss roll. TPE and Isomap (7), a popular manifold learning method, were able to preserve the continuity of the manifold well, while t-SNE artificially created clusters. Details of the experiment can be found in *SI Text*.

Dimensionality reduction methods often have issues with robustness to noise. The dependence of TPE on the SL dendrogram may raise concerns about the sensitivity of the SL dendrogram and TPE to sampling variability. However, the SL dendrogram has been shown to be stable in the sense that small perturbations of the data do not change the structure of the SL dendrogram significantly (30, 41). Therefore, we expect that TPE will also be stable. In order to empirically test the stability of TPE, we simulated 100 samples from a difficult nonconvex manifold, the barbell, and computed the average sample variance of the coordinates of the points in the embeddings. TPE and Isomap had comparable stability to the exact embedding, while t-SNE was two orders of magnitude less stable. Details of the experiment can be found in *SI Text*.

Preserving connectedness allows TPE to preserve different types of structure. However, preserving connectedness is a strong constraint that may not be effective for certain types of structure. Therefore, TPE will not always be able to perform as well as other dimensionality reduction methods in specific applications. For example, manifold learning methods may preserve certain manifolds such as the Swiss roll better and force-based methods may preserve certain clusters such as the handwritten digits better. Nevertheless, we believe that the robustness of TPE is what makes it useful in practice.

TPE is a promising approach to visualization because it has a formal guarantee of cluster separation, requires no parameters, and can handle general types of data. However, there are a few issues with TPE that may limit its applicability. First, TPE has a cubic time complexity, which can be prohibitively slow for large datasets. Second, because TPE only provides an embedding rather than a mapping, it cannot be applied to out-of-sample data. Finally, although we have found that the greedy approximation works well in practice, better optimization methods may significantly improve the performance of TPE. We hope that these issues will be addressed by future research.

ACKNOWLEDGMENTS. An earlier version of this work appeared in the *Proceedings of the 28th International Conference on Machine Learning*. This work was partially supported by the National Science Foundation under grants DMS-0907009 and IIS-1017967, by the National Institutes of Health under Grant R01 GM096193, and by the Army Research Office Multidisciplinary University Research Initiative under Grant 58153-MA-MUR, all to Harvard University. Additional funding was provided by the Harvard Medical School's Milton Fund.

1. Shiffrin RM, Börner K (2004) Mapping knowledge domains. *Proc Natl Acad Sci USA* 101:5183–5185.
2. Hinton GE, Salakhutdinov RR (2006) Reducing the dimensionality of data with neural networks. *Science* 313:504–507.
3. Joliffe IT (2002) *Principal Component Analysis* (Springer, New York).
4. Kruskal JB, Wish M (1978) *Multidimensional Scaling* (Sage Univ Press, Newbury Park).
5. Cox TF, Cox MAA (2001) *Multidimensional Scaling* (Chapman & Hall/CRC, Boca Raton).

6. Borg I, Groenen P (2005) *Modern Multidimensional Scaling: Theory and Applications* (Springer, New York).
7. Tenenbaum JB, de Silva V, Langford JC (2000) A global geometric framework for nonlinear dimensionality reduction. *Science* 290:2319–2323.
8. Roweis ST, Saul LK (2000) Nonlinear dimensionality reduction by locally linear embedding. *Science* 290:2323–2326.
9. Belkin M, Niyogi P (2003) Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput* 15:1373–1396.
10. Venna J, Peltonen J, Nybo K, Aidos H, Samuel K (2010) Information retrieval perspective to nonlinear dimensionality reduction for data visualization. *J Mach Learn Res* 11:451–490.
11. van der Maaten L, Hinton GE (2008) Visualizing data using t-sne. *J Mach Learn Res* 9:2579–2605.
12. Roth V, Laub J, Kawanabe M, Buhmann JM (2003) Optimal cluster preserving embedding of nonmetric proximity data. *IEEE Trans Pattern Anal Mach Intell* 25:1540–1551.
13. Xing EP, Ng AY, Jordan MI, Russell S (2002) Distance metric learning, with application to clustering with side-information. *Adv Neural Inf Process Syst* 14:521–528.
14. Hinton GE, Roweis ST (2003) Stochastic neighbor embedding. *Adv Neural Inf Process Syst* 15:857–864.
15. Cook JA, Sutskever I, Mnih A, Hinton GE (2007) Visualizing similarity data with a mixture of maps. *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics.*
16. Carreira-Perpi nán MA (2010) The elastic embedding algorithm for dimensionality reduction. *Proceedings of the 27th International Conference on Machine Learning* (Omnipress, Haifa, Israel).
17. Chen L, Buja A (2009) Local multidimensional scaling for nonlinear dimension reduction, graph drawing, and proximity analysis. *J Am Stat Assoc* 104:209–219.
18. Di Battista G, Eades P, Tamassia R, Tollis IG (1998) *Graph Drawing: Algorithms for the Visualization of Graphs* (Prentice Hall, New York).
19. Kaufmann M, Wagner D (2001) *Drawing Graphs: Methods and Models* (Springer, New York).
20. Shepard RN (1980) Multidimensional scaling, tree-fitting, and clustering. *Science* 210:390–398.
21. Gower JC, Ross GJS (1969) Minimum spanning trees and single linkage cluster analysis. *Appl Stat* 18:54–64.
22. Zadeh RB, Ben-David S (2009) A uniqueness theorem for clustering. *Proceedings of the 25th Annual Conference on Uncertainty in Artificial Intelligence* (AUAI Press, Montreal).
23. Shaw B, Jebara T (2009) Structure preserving embedding. *Proceedings of the 26th International Conference on Machine Learning* (ACM, Montreal).
24. Eades P (1996) The realization problem for euclidean minimum spanning trees is np-hard. *Algorithmica* 16:60–82.
25. Hartigan JA (1975) *Clustering Algorithms* (Wiley, New York).
26. Hartigan JA (1985) Statistical theory in clustering. *J Classif* 2:63–76.
27. Stuetzle W (2003) Estimating the cluster tree of a density by analyzing the minimal spanning tree of a sample. *J Classif* 20:25–47.
28. Buja A, et al. (2008) Data visualization with multidimensional scaling. *J Comput Graph Stat* 17:444–472.
29. Johnson SC (1967) Hierarchical clustering schemes. *Psychometrika* 32:241–254.
30. Carlsson G, Mémoli F (2010) Characterization, stability and convergence of hierarchical clustering methods. *J Mach Learn Res* 11:1425–1470.
31. Qiu H, Hancock ER (2007) Clustering and embedding using commute times. *IEEE Trans Pattern Anal Mach Intell* 29:1873–1890.
32. Gower JC, Dijksterhuis GB (2004) *Procrustes Problems* (Oxford Univ Press, Oxford).
33. Quist M, Yona G (2004) Distributional scaling: An algorithm for structure-preserving embedding of metric and non-metric spaces. *J Mach Learn Res* 5:399–420.
34. de Silva V, Tenenbaum JB (2003) Global versus local methods for nonlinear dimensionality reduction. *Adv Neural Inf Process Syst* 15:705–712.
35. Pollack JD, Li Q, Pearl DK (2005) Taxonomic utility of a phylogenetic analysis of phosphoglycerate kinase proteins of archaea, bacteria, and eukaryota: Insights by bayesian analyses. *Mol Phylogenet Evol* 35:420–430.
36. Atchley WR, Zhao J, Fernandes AD, Drüke T (2005) Solving the protein sequence metric problem. *Proc Natl Acad Sci USA* 102:6395–6400.
37. Althscul SF, Gish W, Miller W, Myers EW, Lipman DJ (1990) Basic local alignment search tool. *J Mol Biol* 215:403–410.
38. Sigillito VG, Wing SP, Hutton LV, Baker KB (1989) Classification of radar returns from the ionosphere using neural networks. *Johns Hopkins Apl Tech Dig* 10:262–266.
39. Hull JJ (1994) A database for handwritten text recognition research. *IEEE Trans Pattern Anal Mach Intell* 16:550–554.
40. Saul LK, Roweis ST (2003) Think globally, fit locally: Unsupervised learning of low dimensional manifolds. *J Mach Learn Res* 4:119–155.
41. Hartigan JA (1981) Consistency of single linkage for high-density clusters. *J Am Stat Assoc* 76:388–394.

Supporting Information

Shieh et al. 10.1073/pnas.1018393108

SI Text

Greedy Approximation. The greedy approximation must find an alignment of the clusters at each iteration. In practice, a p-dimensional rigid transformation T can be specified by

$$T(x_i) = Rx_i + t$$

where R is a $p \times p$ orthogonal matrix and t is a p-dimensional translation vector. Reflections can be ignored by constraining $\det(R) = 1$ such that R is a proper rotation matrix. Rotation matrices are convenient because a $p \times p$ rotation matrix can be specified by only $p(p-1)/2$ rotation angles. For example, when $p = 2$, a rotation matrix R can be specified by

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

where θ is a rotation angle. For optimization purposes, it is convenient to gather the parameters of a rigid transformation T into a real vector $y(T)$.

At iteration k, we want to find a rigid transformation T^* that minimizes the stress between the clusters

$$\sigma(T) = \sum_{x_i \in X_{a_k}, x_j \in X_{b_k}} (D_{i,j} - d_{i,j}(T))^2,$$

recalling that $d_{i,j}(T) = \|T(x_i) - x_j\|$, subject to the constraint that the minimum nearest neighbor distance between the clusters

$$\Delta(T) = \min_{x_i \in X_{a_k}, x_j \in X_{b_k}} d_{i,j}(T)$$

equals the merge distance Δ_k. That is, we want to solve the optimization problem

$$T^* = \text{argmin}_{T \in E(p)} \sigma(T) \quad \text{s.t.} \quad \Delta(T) = \Delta_k, \qquad \textbf{[S1]}$$

recalling that $E(p)$ is the set of p-dimensional rigid transformations. In practice, satisfying the constraint is often in conflict with minimizing the stress between the clusters due to the crowding problem.

The main difficulty in solving the optimization problem **S1** is satisfying the constraint. Using a quadratic penalty

$$\rho(T) = (\Delta(T) - \Delta_k)^2,$$

we can solve the constrained optimization problem **S1** as an unconstrained optimization problem

$$T^* = \text{argmin}_{T \in E(p)} \sigma(T) + c\rho(T). \qquad \textbf{[S2]}$$

for a sufficiently large value of the penalty weight c. In principle, the optimization problem **S2** can be solved directly. However, for large values of the penalty weight, the optimization problem **S2** becomes highly nonlinear. Therefore, we use a sequential unconstrained minimization technique (1) to gradually increase the penalty weight. The sequential unconstrained minimization technique solves the optimization problem **S2** with a sequence of increasing penalty weights, initializing each problem in the sequence with the solution from the previous problem.

The sequential unconstrained minimization technique proceeds as follows.

1. Set the initialization T^0, the initial penalty weight c, the scale factor λ, and the tolerance ε.
2. Repeat until convergence $\|y(T^0) - y(T^*)\| < \varepsilon$:

i. Solve the optimization problem **S2** with penalty weight c using an unconstrained optimization method initialized from T^0.
ii. Update the initialization $T^0 = T^*$.
iii. Update the penalty weight $c = \lambda c$.

3. Return the rigid transformation T^*.

The sequential unconstrained minimization technique gradually increases the penalty for violating the constraint, allowing a low stress solution that may violate the constraint to be found early on and slowly guided to the feasible region. The sequential unconstrained minimization technique can be started from any initialization, including those outside of the feasible region. For example, using Procrustes analysis to minimize the stress between the clusters without constraints

$$T^0 = \text{argmin}_{T \in E(p)} \sigma(T)$$

provides a reasonable initialization that can be found efficiently (2). The initial penalty weight c, the scale factor λ, and the tolerance ε can be adjusted based on the desired level of speed and accuracy. In practice, reasonable values are an initial penalty weight of $c = 1$, a scale factor of $\lambda = 10$, and a tolerance of $\varepsilon = 10^{-3}$.

Any unconstrained optimization method can be used to solve the optimization problem **S2**. However, there are two difficulties to be aware of. The first difficulty is that the penalty is discontinuous when the nearest neighbors between the clusters change. However, the discontinuities are not severe because the minimum nearest neighbor distance between the clusters is unlikely to change drastically. The second difficulty is that the stress between the clusters is nonconvex. This difficulty applies not only to the greedy approximation but to minimizing stress in general (3). A common strategy to deal with nonconvexity is to use multiple initializations. However, in practice, we found that different initializations often result in similar alignments, suggesting that many local minima provide good alignments.

In practice, we found the use of several heuristics to be helpful. First, using a larger penalty weight for placing the clusters too close together rather than too far apart can improve performance because minimizing the stress between the clusters already tends to drive the clusters close together due to the crowding problem. Second, although an alignment of the clusters is completely specified by a rigid transformation of one of the clusters, allowing for rigid transformations of both clusters can improve performance by providing more degrees of freedom with which to find a good alignment. Finally, reflections are rarely needed to find a good alignment because the orientation of the clusters is unlikely to change when a reasonable initialization is used.

Quantitative Evaluation. In order to quantitatively evaluate the quality of the embeddings, we used several popular performance metrics. First, in order to evaluate the extent to which the embeddings preserve the dissimilarities, we used the normalized stress

$$\sigma(X) = \left(\frac{\sum_{x_i, x_j \in X} (D_{i,j} - d_{i,j}(X))^2}{\sum_{x_i, x_j \in X} D_{i,j}^2} \right)^{1/2},$$

a monotonic transformation of stress that adjusts for the scale of the dissimilarities (3). Second, in order to evaluate the extent to which the embeddings preserve the local neighborhoods, we used the local continuity (4), the fraction of nearest neighbors in the embedding matching the nearest neighbors in the data. Finally, in order to evaluate the extent to which the embeddings preserve the known clusters, we used the clustering coefficient (5), the

fraction of nearest neighbors in the embedding belonging to the same cluster.

Preserving the local neighborhoods and the known clusters, which tend to represent higher-order structure, is widely believed to be more important than preserving the dissimilarities, which tend to represent lower-order structure, for visualization purposes (5). Therefore, performance metrics such as the local continuity and the clustering coefficient are more relevant than performance metrics such as the normalized stress. TPE sacrifices preserving the dissimilarities by preserving connectedness. However, in exchange, the hope is that TPE improves in preserving the local neighborhoods and the known clusters by preserving connectedness.

The normalized stress, local continuity, and clustering coefficient of the embeddings of the protein sequences, radar signals, and handwritten digits are shown in Table S1. Lower values of the normalized stress are better, while higher values of the local continuity and the clustering coefficient are better. TPE and t-SNE both have worse normalized stress than classical methods such as PCA and nonmetric MDS. However, TPE and t-SNE both have better local continuity and clustering coefficient than classical methods. Therefore, TPE and t-SNE both improve on classical methods in the performance metrics more relevant for visualization purposes.

Continuity Test. In order to empirically test whether TPE artificially creates clusters, we generated a sample of the Swiss roll, a popular example of a two-dimensional convex manifold embedded in a three-dimensional ambient space (6, 7). We generated 400 points of the Swiss roll as follows. We generated the manifold (t,x) by sampling the coordinates t uniformly from the interval $[0,2\pi]$ and x uniformly from the interval $[-2,2]$. We generated the ambient space embedding (x,y,z) by sampling noise ε from a normal distribution with mean 0 and standard deviation 0.01 and defining the coordinates $y = (t+\varepsilon)\sin(t^2)$ and $z = (t+\varepsilon)\cos(t^2)$. The Swiss roll is known to contain many holes when sampled sparsely with noise (8), making it a good test for continuity.

We compared TPE to Isomap, a popular manifold learning method known to perform well on the Swiss roll (6), and t-SNE. The embeddings produced by TPE, Isomap, and t-SNE are shown in Fig. S1. Isomap does the best job of recovering the manifold, both preserving continuity and flattening the manifold. TPE preserves continuity well but fails to entirely flatten the manifold. t-SNE fails to either preserve continuity or flatten the manifold and artificially creates clusters instead. TPE is able to preserve continuity because preserving connectedness requires the local neighborhoods of the manifold to remain near each other. We suspect that TPE was unable to flatten the manifold due to the dependence of the greedy approximation on the merge order of the SL dendrogram, which is sensitive to the curvature of the manifold.

Stability Test. In order to empirically test the sensitivity of TPE to sampling variability, we generated 100 samples of the barbell, a popular example of a two-dimensional nonconvex manifold (9). For each sample, we generated 150 points of the barbell as follows. We generated the bells by sampling 50 points each from multivariate normal distributions with means (0,0) and (10,10) and standard deviation 1. We generated the bar by sampling 50 points with coordinates $(u,u)+z$ where u is sampled uniformly from the interval [0,10] and z is sampled from a multivariate normal distribution with mean (0,0) and standard deviation 0.05. The barbell is notoriously difficult to embed due to the presence of both clusters in the bells and continuity in the bar (9), making it a good test for stability of the embedding.

We compared TPE, Isomap, and t-SNE by using Procrustes analysis to align the embeddings produced by TPE, Isomap, and t-SNE to the exact embeddings. The average sample variance of the coordinates of the points in the embeddings was 3.45 for the exact embeddings, 2.72 for TPE, 3.40 for Isomap, and 984.54 for t-SNE. The embeddings produced by TPE and Isomap both had comparable stability to the exact embeddings, while the embeddings produced by t-SNE were two orders of magnitude more variable. The embeddings of two samples of the barbell produced by TPE, Isomap, and t-SNE are shown in Fig. S2. Only TPE is able to preserve the shapes of the bar and the bells. Isomap preserves the shape of the bar, but arbitrarily collapses the shape of one of the bells. t-SNE fails to preserve the shapes of either the bar or the bells.

Proofs.

Proof of Theorem 1: The greedy approximation provides an explicit construction of an embedding. Without loss of generality, consider the greedy approximation for a one-dimensional embedding such that the clusters are contained within intervals. At each iteration, ignoring the minimization of the stress between the clusters, we can always place one of the clusters to the right or left of the other cluster by exactly their merge distance apart to satisfy the connectedness constraints. Therefore, the greedy approximation always provides a feasible solution.

Proof of Theorem 2: First, we show that if objects $i, j \in S$ are ε-connected, then points $x_i, x_j \in X$ are ε-connected. We know that x_i, x_j are δ-connected by the distance $\delta = \Delta_k$ at which they are merged into the same cluster at some iteration k. Let $\alpha_1 = i,\ldots,\alpha_m = j \in S$ be the path of objects that ε-connects i, j. Because $i \in S_{a_k}$ and $j \in S_{b_k}$, there exists some l such that $\alpha_l \in S_{a_k}$ and $\alpha_{l+1} \in S_{b_k}$. Because α_l and α_{l+1} are in different clusters, $\delta \le D_{\alpha_l,\alpha_{l+1}} \le \varepsilon$. Therefore, x_i, x_j are ε-connected.

Now, we show that if points $x_i, x_j \in X$ are ε-connected, then objects $i, j \in S$ are ε-connected. Without loss of generality, let x_i, x_j be points merged into the same cluster at iteration k such that they are ε-connected by the distance $\varepsilon = \Delta_k$. Let $x_{\alpha_1} = x_i,\ldots x_{\alpha_m} = x_j \in X$ be the path of points that ε-connects x_i, x_j. We know that there exists some l such that $d_{\alpha_l,\alpha_{l+1}}(X) = \Delta_k$. Let $k_l \le k$ be the minimum number of iterations such that $\alpha_1,\ldots,\alpha_l$ are in the same cluster. Then $\alpha_1,\ldots,\alpha_l$ are δ-connected by $\delta = \Delta_{k_l}$. Since Δ_k is monotonically increasing in k, $\delta \le \varepsilon$. Therefore, $\alpha_1,\ldots,\alpha_l$ are ε-connected. Similarly, $\alpha_{l+1},\ldots,\alpha_m$ are ε-connected. Since α_l and α_{l+1} are in different clusters, $D_{\alpha_l,\alpha_{l+1}} \le \varepsilon$. Therefore, i, j are ε-connected.

Proof of Theorem 3: The lower bound follows directly from the connectedness constraints. The upper bound follows from

$$d_{ij}(X) \le \max_{\alpha_1=i,\ldots,\alpha_m=j\in S_{n+k}} \sum_{l=1}^{m-1} d_{\alpha_l,\alpha_{l+1}}(X)$$

$$\le \max_{i',j'\in S_{n+k}} \max_{\alpha_1=i',\ldots,\alpha_m=j'\in S_{n+k}} \sum_{l=1}^{m-1} d_{\alpha_l,\alpha_{l+1}}(X) = U_{ij}$$

where we have assumed that the paths have no cycles and used the triangle inequality and the fact that the hops of the paths cannot exceed the merge distances. The bounds are illustrated in Fig. S3.

Proof of Corollary 4: By Theorem 3, we have

$$(d_{ij}(X) - D_{ij})^2 \le \max\{(D_{ij} - L_{ij})^2, (D_{ij} - U_{ij})^2\}$$

Summing over all $x_i, x_j \in X$, we have

$$\sigma(X) \le \sum_{x_i,x_j\in X} \max\{(D_{ij} - L_{ij})^2, (D_{ij} - U_{ij})^2\}.$$

1. Nocedal J, Wright S (2006) *Numerical Optimization* Springer, New York.
2. Gower JC, Dijksterhuis GB (2004) *Procrustes Problems* Oxford Univ Press, Oxford.
3. Buja A, et al. (2008) Data visualization with multidimensional scaling. *J Comput Graph Stat* 17:444–472.
4. Chen L, Buja A (2009) Local multidimensional scaling for nonlinear dimension reduction, graph drawing, and proximity analysis. *J Am Stat Assoc* 104:209–219.
5. Venna J, Peltonen J, Nybo K, Aidos H, Samuel K (2010) Information retrieval perspective to nonlinear dimensionality reduction for data visualization. *J Mach Learn Res* 11:451–490.
6. Tenenbaum JB, de Silva V, Langford JC (2000) A global geometric framework for nonlinear dimensionality reduction. *Science* 290:2319–2323.
7. Roweis ST, Saul LK (2000) Nonlinear dimensionality reduction by locally linear embedding. *Science* 290:2323–2326.
8. Balasubramanian M, Schwartz EL, Tenenbaum JB, de Silva V, Langford JC (2002) The isomap algorithm and topological stability. *Science* 295:7.
9. Saul LK, Roweis ST (2003) Think globally, fit locally: Unsupervised learning of low dimensional manifolds. *J Mach Learn Res* 4:119–155.

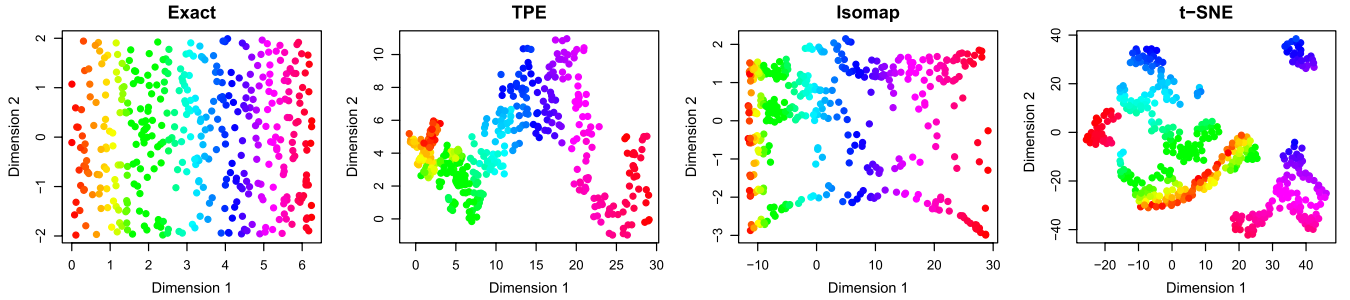

Fig. S1. Embeddings of the Swiss roll produced by TPE, Isomap, and t-SNE.

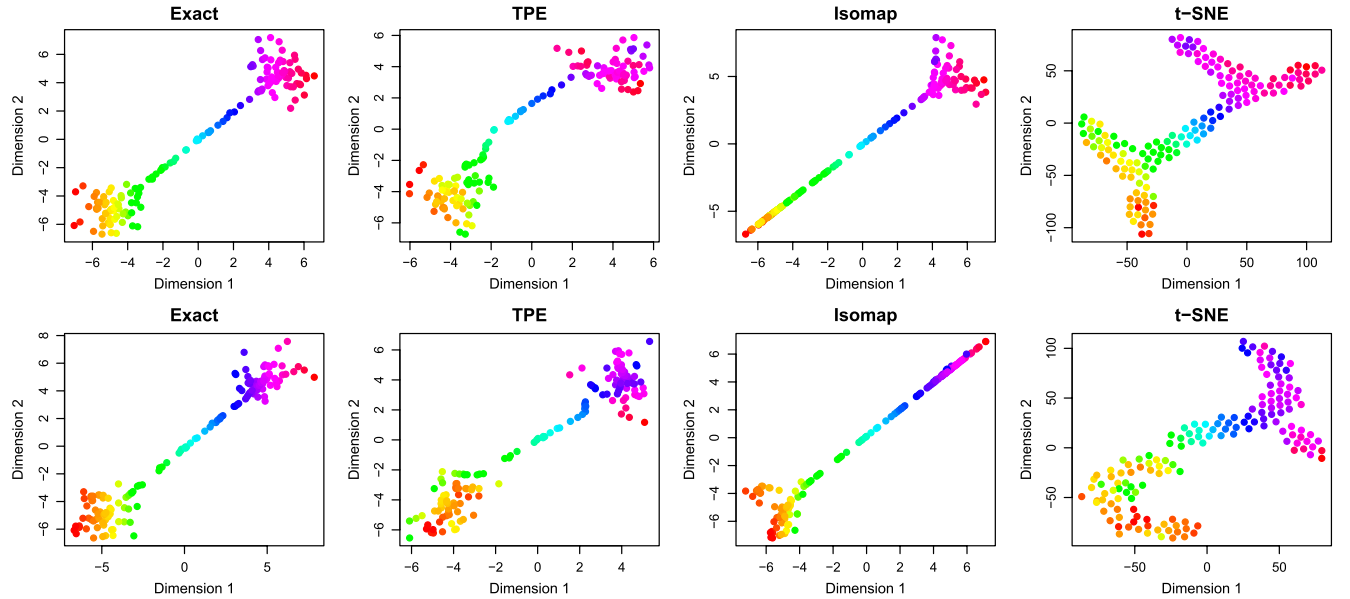

Fig. S2. Embeddings of two samples of the barbell produced by TPE, Isomap, and t-SNE.

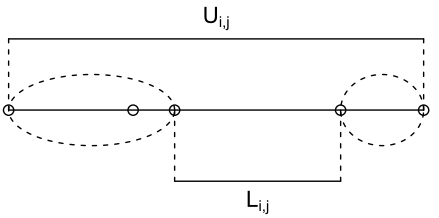

Fig. S3. Bounds on the Euclidean distance between points merged into the same cluster at iteration k. The ellipses denote the clusters being merged at iteration k, $L_{i,j}$ denotes the lower bound, and $U_{i,j}$ denotes the upper bound.

Table S1. Normalized stress (NS), local continuity (LC), and clustering coefficient (CC) of embeddings of protein sequences, radar signals, and handwritten digits

Example	Method	NS	LC	CC
Protein Sequences	TPE	**6.619**	**0.611**	**0.992**
	Nonmetric MDS	30.658	0.198	0.641
	t-SNE	456.375	0.206	0.855
Radar Signals	TPE	2.187	**0.365**	**0.923**
	PCA	**0.453**	0.205	0.732
	t-SNE	15.521	0.305	0.880
Handwritten Digits	TPE	8.322	**0.627**	0.867
	PCA	**0.558**	0.030	0.529
	t-SNE	2.070	0.473	**0.931**

Lower values of NS are better, while higher values of LC and CC are better.