35. R. N. Shepard, *Psychon. Bull. Rev.* **1**, 2 (1994).
36. J. B. Tenenbaum, *Adv. Neural Info. Proc. Syst.* **10**, 682 (1998).
37. T. Martinetz, K. Schulten, *Neural Netw.* **7**, 507 (1994).
38. V. Kumar, A. Grama, A. Gupta, G. Karypis, *Introduction to Parallel Computing: Design and Analysis of Algorithms* (Benjamin/Cummings, Redwood City, CA, 1994), pp. 257–297.
39. D. Beymer, T. Poggio, *Science* **272**, 1905 (1996).
40. Available at www.research.att.com/~yann/ocr/mnist.
41. P. Y. Simard, Y. LeCun, J. Denker, *Adv. Neural Info. Proc. Syst.* **5**, 50 (1993).
42. In order to evaluate the fits of PCA, MDS, and Isomap on comparable grounds, we use the residual variance $1 - R^2(\hat{D}_M, D_Y)$. D_Y is the matrix of Euclidean distances in the low-dimensional embedding recovered by each algorithm. $\hat{D}_M$ is each algorithm's best estimate of the intrinsic manifold distances: for Isomap, this is the graph distance matrix D_G; for PCA and MDS, it is the Euclidean input-space distance matrix D_X (except with the handwritten "2"s, where MDS uses the tangent distance). R is the standard linear correlation coefficient, taken over all entries of $\hat{D}_M$ and D_Y.
43. In each sequence shown, the three intermediate images are those closest to the points 1/4, 1/2, and 3/4 of the way between the given endpoints. We can also synthesize an explicit mapping from input space X to the low-dimensional embedding Y, or vice versa, using the coordinates of corresponding points $\{\mathbf{x}_i, \mathbf{y}_i\}$ in both spaces provided by Isomap together with standard supervised learning techniques (*39*).
44. Supported by the Mitsubishi Electric Research Laboratories, the Schlumberger Foundation, the NSF (DBS-9021648), and the DARPA Human ID program. We thank Y. LeCun for making available the MNIST database and S. Roweis and L. Saul for sharing related unpublished work. For many helpful discussions, we thank G. Carlsson, H. Farid, W. Freeman, T. Griffiths, R. Lehrer, S. Mahajan, D. Reich, W. Richards, J. M. Tenenbaum, Y. Weiss, and especially M. Bernstein.

10 August 2000; accepted 21 November 2000

Nonlinear Dimensionality Reduction by Locally Linear Embedding

Sam T. Roweis[1] and Lawrence K. Saul[2]

Many areas of science depend on exploratory data analysis and visualization. The need to analyze large amounts of multivariate data raises the fundamental problem of dimensionality reduction: how to discover compact representations of high-dimensional data. Here, we introduce locally linear embedding (LLE), an unsupervised learning algorithm that computes low-dimensional, neighborhood-preserving embeddings of high-dimensional inputs. Unlike clustering methods for local dimensionality reduction, LLE maps its inputs into a single global coordinate system of lower dimensionality, and its optimizations do not involve local minima. By exploiting the local symmetries of linear reconstructions, LLE is able to learn the global structure of nonlinear manifolds, such as those generated by images of faces or documents of text.

How do we judge similarity? Our mental representations of the world are formed by processing large numbers of sensory inputs—including, for example, the pixel intensities of images, the power spectra of sounds, and the joint angles of articulated bodies. While complex stimuli of this form can be represented by points in a high-dimensional vector space, they typically have a much more compact description. Coherent structure in the world leads to strong correlations between inputs (such as between neighboring pixels in images), generating observations that lie on or close to a smooth low-dimensional manifold. To compare and classify such observations—in effect, to reason about the world—depends crucially on modeling the nonlinear geometry of these low-dimensional manifolds.

Scientists interested in exploratory analysis or visualization of multivariate data (*1*) face a similar problem in dimensionality reduction. The problem, as illustrated in Fig. 1, involves mapping high-dimensional inputs into a low-dimensional "description" space with as many coordinates as observed modes of variability. Previous approaches to this problem, based on multidimensional scaling (MDS) (*2*), have computed embeddings that attempt to preserve pairwise distances [or generalized disparities (*3*)] between data points; these distances are measured along straight lines or, in more sophisticated usages of MDS such as Isomap (*4*), along shortest paths confined to the manifold of observed inputs. Here, we take a different approach, called locally linear embedding (LLE), that eliminates the need to estimate pairwise distances between widely separated data points. Unlike previous methods, LLE recovers global nonlinear structure from locally linear fits.

The LLE algorithm, summarized in Fig. 2, is based on simple geometric intuitions. Suppose the data consist of N real-valued vectors $\vec{X}_i$, each of dimensionality D, sampled from some underlying manifold. Provided there is sufficient data (such that the manifold is well-sampled), we expect each data point and its neighbors to lie on or close to a locally linear patch of the manifold. We characterize the local geometry of these patches by linear coefficients that reconstruct each data point from its neighbors. Reconstruction errors are measured by the cost function

$$\varepsilon(W) = \sum_i \left| \vec{X}_i - \sum_j W_{ij} \vec{X}_j \right|^2 \qquad (1)$$

which adds up the squared distances between all the data points and their reconstructions. The weights W_{ij} summarize the contribution of the jth data point to the ith reconstruction. To compute the weights W_{ij}, we minimize the cost

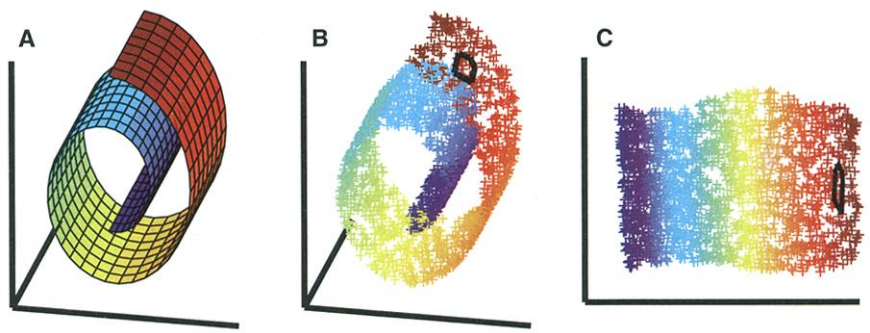

Fig. 1. The problem of nonlinear dimensionality reduction, as illustrated (*10*) for three-dimensional data (B) sampled from a two-dimensional manifold (**A**). An unsupervised learning algorithm must discover the global internal coordinates of the manifold without signals that explicitly indicate how the data should be embedded in two dimensions. The color coding illustrates the neighborhood-preserving mapping discovered by LLE; black outlines in (**B**) and (**C**) show the neighborhood of a single point. Unlike LLE, projections of the data by principal component analysis (PCA) (*28*) or classical MDS (*2*) map faraway data points to nearby points in the plane, failing to identify the underlying structure of the manifold. Note that mixture models for local dimensionality reduction (*29*), which cluster the data and perform PCA within each cluster, do not address the problem considered here: namely, how to map high-dimensional data into a single global coordinate system of lower dimensionality.

[1]Gatsby Computational Neuroscience Unit, University College London, 17 Queen Square, London WC1N 3AR, UK. [2]AT&T Lab—Research, 180 Park Avenue, Florham Park, NJ 07932, USA.

E-mail: roweis@gatsby.ucl.ac.uk (S.T.R.); lsaul@research.att.com (L.K.S.)

function subject to two constraints: first, that each data point $\vec{X}_i$ is reconstructed only from its neighbors (5), enforcing $W_{ij} = 0$ if $\vec{X}_j$ does not belong to the set of neighbors of $\vec{X}_i$; second, that the rows of the weight matrix sum to one: $\Sigma_j W_{ij} = 1$. The optimal weights W_{ij} subject to these constraints (6) are found by solving a least-squares problem (7).

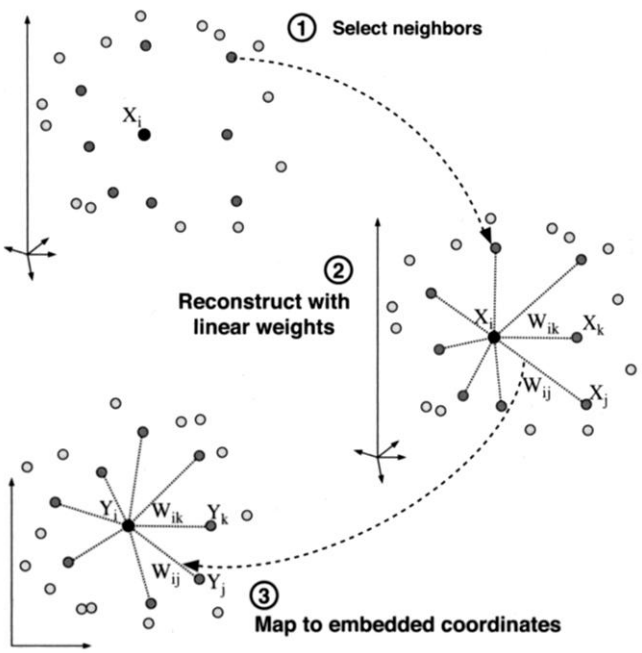

Fig. 2. Steps of locally linear embedding: (1) Assign neighbors to each data point $\vec{X}_i$ (for example by using the K nearest neighbors). (2) Compute the weights W_{ij} that best linearly reconstruct $\vec{X}_i$ from its neighbors, solving the constrained least-squares problem in Eq. 1. (3) Compute the low-dimensional embedding vectors $\vec{Y}_i$ best reconstructed by W_{ij}, minimizing Eq. 2 by finding the smallest eigenmodes of the sparse symmetric matrix in Eq. 3. Although the weights W_{ij} and vectors Y_i are computed by methods in linear algebra, the constraint that points are only reconstructed from neighbors can result in highly nonlinear embeddings.

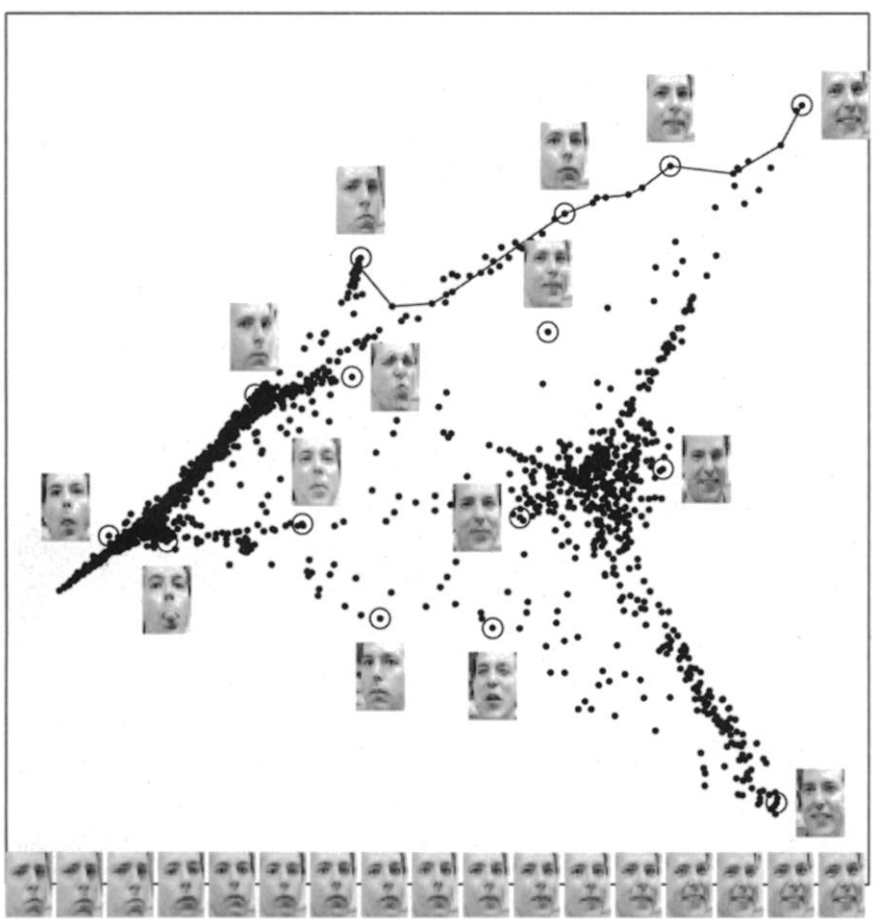

Fig. 3. Images of faces (11) mapped into the embedding space described by the first two coordinates of LLE. Representative faces are shown next to circled points in different parts of the space. The bottom images correspond to points along the top-right path (linked by solid line), illustrating one particular mode of variability in pose and expression.

The constrained weights that minimize these reconstruction errors obey an important symmetry: for any particular data point, they are invariant to rotations, rescalings, and translations of that data point and its neighbors. By symmetry, it follows that the reconstruction weights characterize intrinsic geometric properties of each neighborhood, as opposed to properties that depend on a particular frame of reference (8). Note that the invariance to translations is specifically enforced by the sum-to-one constraint on the rows of the weight matrix.

Suppose the data lie on or near a smooth nonlinear manifold of lower dimensionality $d \ll D$. To a good approximation then, there exists a linear mapping—consisting of a translation, rotation, and rescaling—that maps the high-dimensional coordinates of each neighborhood to global internal coordinates on the manifold. By design, the reconstruction weights W_{ij} reflect intrinsic geometric properties of the data that are invariant to exactly such transformations. We therefore expect their characterization of local geometry in the original data space to be equally valid for local patches on the manifold. In particular, the same weights W_{ij} that reconstruct the ith data point in D dimensions should also reconstruct its embedded manifold coordinates in d dimensions.

LLE constructs a neighborhood-preserving mapping based on the above idea. In the final step of the algorithm, each high-dimensional observation $\vec{X}_i$ is mapped to a low-dimensional vector $\vec{Y}_i$ representing global internal coordinates on the manifold. This is done by choosing d-dimensional coordinates $\vec{Y}_i$ to minimize the embedding cost function

$$\Phi(Y) = \sum_i \left| \vec{Y}_i - \Sigma_j W_{ij} \vec{Y}_j \right|^2 \qquad (2)$$

This cost function, like the previous one, is based on locally linear reconstruction errors, but here we fix the weights W_{ij} while optimizing the coordinates $\vec{Y}_i$. The embedding cost in Eq. 2 defines a quadratic form in the vectors $\vec{Y}_i$. Subject to constraints that make the problem well-posed, it can be minimized by solving a sparse $N \times N$ eigenvalue problem (9), whose bottom d nonzero eigenvectors provide an ordered set of orthogonal coordinates centered on the origin.

Implementation of the algorithm is straightforward. In our experiments, data points were reconstructed from their K nearest neighbors, as measured by Euclidean distance or normalized dot products. For such implementations of LLE, the algorithm has only one free parameter: the number of neighbors, K. Once neighbors are chosen, the optimal weights W_{ij} and coordinates $\vec{Y}_i$ are

computed by standard methods in linear algebra. The algorithm involves a single pass through the three steps in Fig. 2 and finds global minima of the reconstruction and embedding costs in Eqs. 1 and 2.

In addition to the example in Fig. 1, for which the true manifold structure was known (*10*), we also applied LLE to images of faces (*11*) and vectors of word-document counts (*12*). Two-dimensional embeddings of faces and words are shown in Figs. 3 and 4. Note how the coordinates of these embedding spaces are related to meaningful attributes, such as the pose and expression of human faces and the semantic associations of words.

Many popular learning algorithms for nonlinear dimensionality reduction do not share the favorable properties of LLE. Iterative hill-climbing methods for autoencoder neural networks (*13, 14*), self-organizing maps (*15*), and latent variable models (*16*) do not have the same guarantees of global optimality or convergence; they also tend to involve many more free parameters, such as learning rates, convergence criteria, and architectural specifications. Finally, whereas other nonlinear methods rely on deterministic annealing schemes (*17*) to avoid local minima, the optimizations of LLE are especially tractable.

LLE scales well with the intrinsic manifold dimensionality, d, and does not require a discretized gridding of the embedding space. As more dimensions are added to the embedding space, the existing ones do not change, so that LLE does not have to be rerun to compute higher dimensional embeddings. Unlike methods such as principal curves and surfaces (*18*) or additive component models (*19*), LLE is not limited in practice to manifolds of extremely low dimensionality or codimensionality. Also, the intrinsic value of d can itself be estimated by analyzing a reciprocal cost function, in which reconstruction weights derived from the embedding vectors $\vec{Y}_i$ are applied to the data points $\vec{X}_i$.

LLE illustrates a general principle of manifold learning, elucidated by Martinetz and Schulten (*20*) and Tenenbaum (*4*), that overlapping local neighborhoods—collectively analyzed—can provide information about global geometry. Many virtues of LLE are shared by Tenenbaum's algorithm, Isomap, which has been successfully applied to similar problems in nonlinear dimensionality reduction. Isomap's embeddings, however, are optimized to preserve geodesic distances between general pairs of data points, which can only be estimated by computing shortest paths through large sublattices of data. LLE takes a different approach, analyzing local symmetries, linear coefficients, and reconstruction errors instead of global constraints, pairwise distances, and stress functions. It thus avoids the need to solve large dynamic programming problems, and it also tends to accumulate very sparse matrices, whose structure can be exploited for savings in time and space.

LLE is likely to be even more useful in combination with other methods in data analysis and statistical learning. For example, a parametric mapping between the observation and embedding spaces could be learned by supervised neural networks (*21*) whose target values are generated by LLE. LLE can also be generalized to harder settings, such as the case of disjoint data manifolds (*22*), and specialized to simpler ones, such as the case of time-ordered observations (*23*).

Perhaps the greatest potential lies in applying LLE to diverse problems beyond those considered here. Given the broad appeal of traditional methods, such as PCA and MDS, the algorithm should find widespread use in many areas of science.

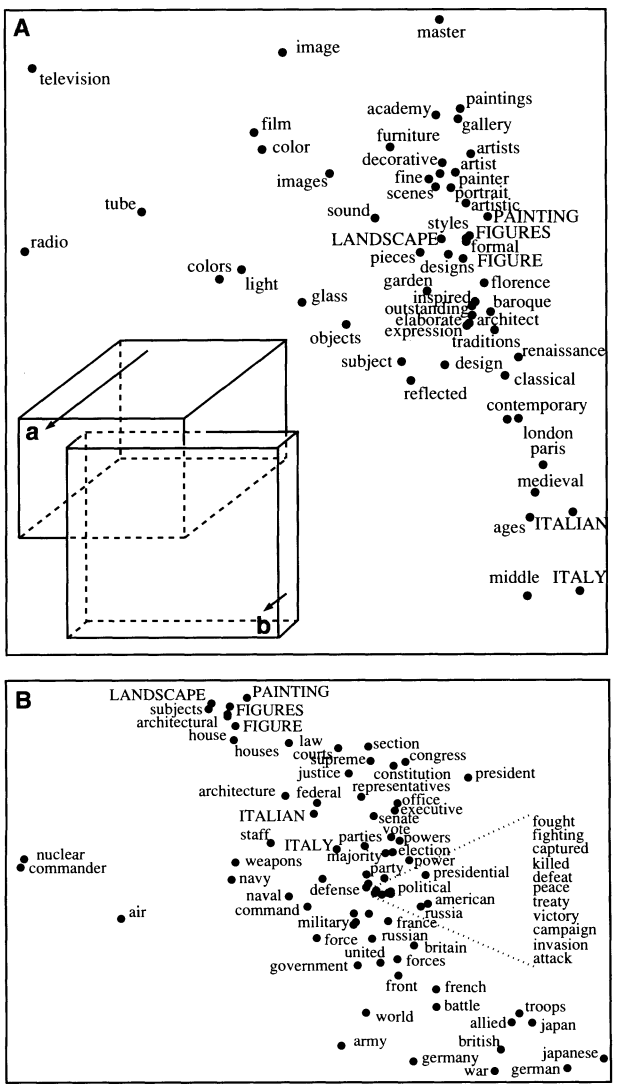

Fig. 4. Arranging words in a continuous semantic space. Each word was initially represented by a high-dimensional vector that counted the number of times it appeared in different encyclopedia articles. LLE was applied to these word-document count vectors (*12*), resulting in an embedding location for each word. Shown are words from two different bounded regions (**A**) and (**B**) of the embedding space discovered by LLE. Each panel shows a two-dimensional projection onto the third and fourth coordinates of LLE; in these two dimensions, the regions (A) and (B) are highly overlapped. The inset in (A) shows a three-dimensional projection onto the third, fourth, and fifth coordinates, revealing an extra dimension along which regions (A) and (B) are more separated. Words that lie in the intersection of both regions are capitalized. Note how LLE colocates words with similar contexts in this continuous semantic space.

References and Notes

1. M. L. Littman, D. F. Swayne, N. Dean, A. Buja, in *Computing Science and Statistics: Proceedings of the 24th Symposium on the Interface*, H. J. N. Newton, Ed. (Interface Foundation of North America, Fairfax Station, VA, 1992), pp. 208–217.
2. T. Cox, M. Cox, *Multidimensional Scaling* (Chapman & Hall, London, 1994).
3. Y. Takane, F. W. Young, *Psychometrika* **42**, 7 (1977).
4. J. Tenenbaum, in *Advances in Neural Information Processing 10*, M. Jordan, M. Kearns, S. Solla, Eds. (MIT Press, Cambridge, MA, 1998), pp. 682–688.
5. The set of neighbors for each data point can be assigned in a variety of ways: by choosing the K nearest neighbors in Euclidean distance, by considering all data points within a ball of fixed radius, or by using prior knowledge. Note that for fixed number of neighbors, the maximum number of embedding dimensions LLE can be expected to recover is strictly less than the number of neighbors.
6. For certain applications, one might also constrain the weights to be positive, thus requiring the reconstruction of each data point to lie within the convex hull of its neighbors.
7. Fits: The constrained weights that best reconstruct each data point from its neighbors can be computed in closed form. Consider a particular data point $\vec{x}$ with neighbors $\vec{\eta}_j$ and sum-to-one reconstruction weights w_j. The reconstruction error $|\vec{x} - \sum_{j=1}^{K} w_j\vec{\eta}_j|^2$ is minimized in three steps. First, evaluate inner products between neighbors to compute the neighborhood correlation matrix, $C_{jk} = \vec{\eta}_j \cdot \vec{\eta}_k$, and its matrix inverse, C^{-1}. Second, compute the Lagrange multiplier, $\lambda = \alpha/\beta$, that enforces the sum-to-one constraint, where $\alpha = 1 - \sum_{jk} C_{jk}^{-1}(\vec{x} \cdot \vec{\eta}_k)$ and $\beta = \sum_{jk} C_{jk}^{-1}$. Third, compute the reconstruction weights: $w_j = \sum_k C_{jk}^{-1}(\vec{x} \cdot \vec{\eta}_k + \lambda)$. If the correlation matrix C is

nearly singular, it can be conditioned (before inversion) by adding a small multiple of the identity matrix. This amounts to penalizing large weights that exploit correlations beyond some level of precision in the data sampling process.

8. Indeed, LLE does not require the original data to be described in a single coordinate system, only that each data point be located in relation to its neighbors.
9. The embedding vectors $\vec{Y}_i$ are found by minimizing the cost function $\Phi(Y) = \Sigma_i |\vec{Y}_i - \Sigma_j W_{ij}\vec{Y}_j|^2$ over $\vec{Y}_i$ with fixed weights W_{ij}. This optimization is performed subject to constraints that make the problem well posed. It is clear that the coordinates $\vec{Y}_i$ can be translated by a constant displacement without affecting the cost, $\Phi(Y)$. We remove this degree of freedom by requiring the coordinates to be centered on the origin: $\Sigma_i \vec{Y}_i = \vec{0}$. Also, to avoid degenerate solutions, we constrain the embedding vectors to have unit covariance, with outer products that satisfy $\frac{1}{N}\Sigma_i \vec{Y}_i \otimes \vec{Y}_i = I$, where I is the $d \times d$ identity matrix. Now the cost defines a quadratic form, $\Phi(Y) = \Sigma_{ij} M_{ij}(\vec{Y}_i \cdot \vec{Y}_j)$, involving inner products of the embedding vectors and the symmetric $N \times N$ matrix

$$M_{ij} = \delta_{ij} - W_{ij} - W_{ji} + \sum_k W_{ki}W_{kj} \quad (3)$$

where δ_{ij} is 1 if $i = j$ and 0 otherwise. The optimal embedding, up to a global rotation of the embedding space, is found by computing the bottom $d + 1$ eigenvectors of this matrix (*24*). The bottom eigenvector of this matrix, which we discard, is the unit vector with all equal components; it represents a free translation mode of eigenvalue zero. (Discarding it enforces the constraint that the embeddings have zero mean.) The remaining d eigenvectors form the d embedding coordinates found by LLE. Note that the matrix M can be stored and manipulated as the sparse matrix $(I - W)^T(I - W)$, giving substantial computational savings for large values of N. Moreover, its bottom $d + 1$ eigenvectors (those corresponding to its smallest $d + 1$ eigenvalues) can be found efficiently without performing a full matrix diagonalization (*25*).
10. Manifold: Data points in Fig. 1B ($N = 2000$) were sampled from the manifold ($D = 3$) shown in Fig. 1A. Nearest neighbors ($K = 20$) were determined by Euclidean distance. This particular manifold was introduced by Tenenbaum (*4*), who showed that its global structure could be learned by the Isomap algorithm.
11. Faces: Multiple photographs ($N = 2000$) of the same face were digitized as 20×28 grayscale images. Each image was treated by LLE as a data vector with $D = 560$ elements corresponding to raw pixel intensities. Nearest neighbors ($K = 12$) were determined by Euclidean distance in pixel space.
12. Words: Word-document counts were tabulated for $N = 5000$ words from $D = 31{,}000$ articles in Grolier's Encyclopedia (*26*). Nearest neighbors ($K = 20$) were determined by dot products between count vectors normalized to unit length.
13. D. DeMers, G. W. Cottrell, in *Advances in Neural Information Processing Systems 5*, D. Hanson, J. Cowan, L. Giles, Eds. (Kaufmann, San Mateo, CA, 1993), pp. 580–587.
14. M. Kramer, *AIChE J.* **37**, 233 (1991).
15. T. Kohonen, *Self-Organization and Associative Memory* (Springer-Verlag, Berlin, 1988).
16. C. Bishop, M. Svensen, C. Williams, *Neural Comput.* **10**, 215 (1998).
17. H. Klock, J. Buhmann, *Pattern Recognition* **33**, 651 (1999).
18. T. J. Hastie, W. Stuetzle, *J. Am. Stat. Assoc.* **84**, 502 (1989).
19. D. J. Donnell, A. Buja, W. Stuetzle, *Ann. Stat.* **22**, 1635 (1994).
20. T. Martinetz, K. Schulten, *Neural Networks* **7**, 507 (1994).
21. D. Beymer, T. Poggio, *Science* **272**, 1905 (1996).
22. Although in all the examples considered here, the data had a single connected component, it is possible to formulate LLE for data that lies on several disjoint manifolds, possibly of different underlying dimensionality. Suppose we form a graph by connecting each data point to its neighbors. The number of connected components (*27*) can be detected by examining powers of its adjacency matrix. Different connected components of the data are essentially decoupled in the eigenvector problem for LLE. Thus, they are best interpreted as lying on distinct manifolds, and are best analyzed separately by LLE.
23. If neighbors correspond to nearby observations in time, then the reconstruction weights can be computed online (as the data itself is being collected) and the embedding can be found by diagonalizing a sparse banded matrix.
24. R. A. Horn, C. R. Johnson, *Matrix Analysis* (Cambridge Univ. Press, Cambridge, 1990).
25. Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, H. van der Vorst, Eds., *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide* (Society for Industrial and Applied Mathematics, Philadelphia, PA, 2000).
26. D. D. Lee, H. S. Seung, *Nature* **401**, 788 (1999).
27. R. Tarjan, *Data Structures and Network Algorithms, CBMS 44* (Society for Industrial and Applied Mathematics, Philadelphia, PA, 1983).
28. I. T. Jolliffe, *Principal Component Analysis* (Springer-Verlag, New York, 1989).
29. N. Kambhatla, T. K. Leen, *Neural Comput.* **9**, 1493 (1997).
30. We thank G. Hinton and M. Revow for sharing their unpublished work (at the University of Toronto) on segmentation and pose estimation that motivated us to "think globally, fit locally"; J. Tenenbaum (Stanford University) for many stimulating discussions about his work (*4*) and for sharing his code for the Isomap algorithm; D. D. Lee (Bell Labs) and B. Frey (University of Waterloo) for making available word and face data from previous work (*26*); and C. Brody, A. Buja, P. Dayan, Z. Ghahramani, G. Hinton, T. Jaakkola, D. Lee, F. Pereira, and M. Sahani for helpful comments. S.T.R. acknowledges the support of the Gatsby Charitable Foundation, the U.S. National Science Foundation, and the National Sciences and Engineering Research Council of Canada.

7 August 2000; accepted 17 November 2000