

# Manifold Learning for Visualizing and Analyzing High-Dimensional Data

Junping Zhang, *Fudan University*

Hua Huang, *Xi'an Jiaotong University*

Jue Wang, *Chinese Academy of Sciences*

**Assuming that high-dimensional data are generated from intrinsic variables with lower dimensions, several key manifold-learning algorithms can help effectively analyze and visualize such data.**

**H**andling large-scale, high-dimensional data is a challenging task in modern statistical data analysis. Such data have become more readily available as our collection capacity vastly improves. Examples of such data include high-resolution images and videos, sensor network readings, and genomics.

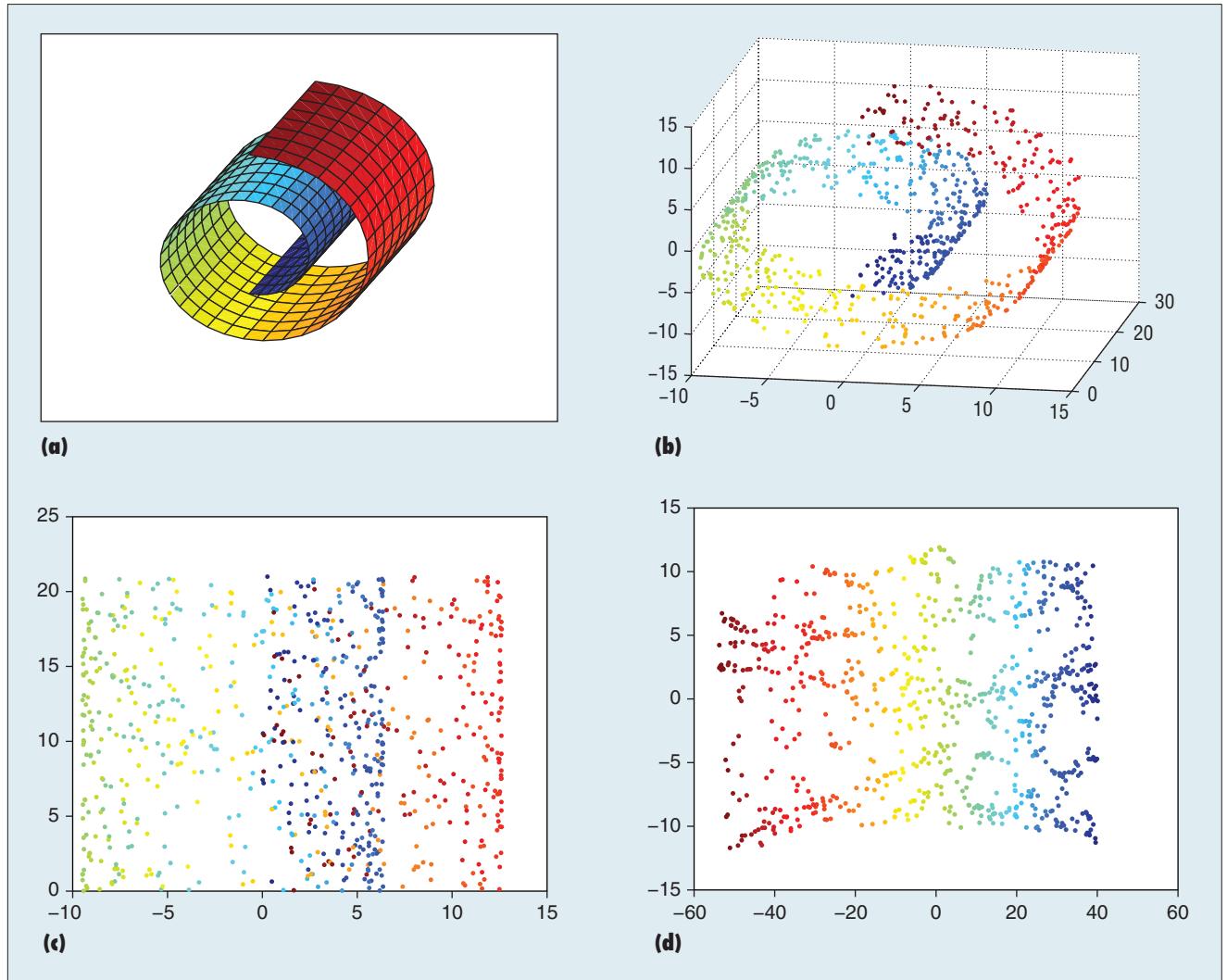
In a simplistic representation, many of these data are represented as points in high-dimensional Euclidean spaces. For instance, we can represent an image as a vector of pixel intensities after stacking up all the pixels according to some ordering. The result is a high-dimensional vector—its dimensionality is the number of pixels in the image.

It is difficult to gain insight from such data, however, because conventional approaches such as visualization and exploratory data analysis do not scale well to more than a few dozen dimensions. Furthermore, statistically modeling these data often encounters the *curse of dimensionality*, in which the amount of data needed for robust statistical modeling grows exponentially in the dimensionality. For these reasons, dimension reduction has become an imperative processing step for visualizing and analyzing high-dimensional data.

This article describes a few representative manifold-learning algorithms and demonstrates their utility in visualizing and analyzing high-dimensional data. We also discuss their strengths and weaknesses, along with strategies to avoid pitfalls.

## Linear Projection versus Manifold Learning

Researchers have extensively studied dimension reduction in many fields, including statistics, machine learning, and data mining. A classical example is principal component analysis (PCA), which assumes that we can characterize data with coordinates in linear subspaces with lower dimensionality. The coordinates can be computed as a linear projection from the original high-dimensional space so that we can use them to optimally reconstruct the original data points. (Optimal reconstruction is just one



**Figure 1. Unfolding 800 3D Swiss-roll data points to a 2D space with principal component analysis (PCA) and Isomap algorithms.**  
**(a) Swiss roll, (b) Swiss roll data, (c) PCA, and (d) Isomap.**

of the many approaches to derive the technique of PCA.) PCA is especially computationally tractable and scales to both large-scale and high-dimensional data sets. The core step is to diagonalize the data's covariance matrix. Alternatively, we can diagonalize the Gram matrix, with elements being the pairwise inner products between data points.

Despite its popularity and wide application, linear projection (including PCA) is still inadequate in capturing interesting structures in data if the data do not live in a linear subspace. Figure 1a illustrates an example

by depicting a Swiss-roll object in 3D space. This object can be identified as a 2D rectangle being rolled up. Therefore, it is inherently 2D and can be characterized with two coordinates. If we have data points sampled from this object, as Figure 1b shows, how can an algorithm (for dimension reduction) discover the 2D coordinates of each sampled point?

Figure 1c illustrates the results computed by PCA. The two ends of the Swiss roll, in red and blue, respectively, are mixed on the 2D plane. These two ends are far away in the Swiss roll's 2D intrinsic structure

because they are on opposite sides of the rectangle. However, PCA misidentifies them as being close and projects them in 2D as nearby points. In fact, any linear projection method would fail to discover the correct 2D structure.

In contrast, the Isomap algorithm—a manifold-learning technique—correctly identifies the intrinsic structure in Figure 1d. The power of Isomap and other similar approaches relies on the key notion of a manifold, which is a smooth object that locally looks like a Euclidean space, yet globally has a

complicated and nonlinear structure. In other words, we can approximate complicated structures by gluing small linear subspaces and structures together. Manifold-learning algorithms infer global structures from locally computed geometric properties (such as distances, angles, and symmetry). Additionally, such inferences can be made computationally very appealing. While implementing nonlinear projection, many of the manifold-learning algorithms depend on similar computational steps as in linear methods, such as matrix diagonalization. In most cases, they are only slightly more computationally intensive than linear methods.

The existence of intrinsic structures such as manifolds is often motivated by several related schools of thought. A small number of latent variables can help explain high-dimensional data. For instance, we can consider distributions (relative occurrence frequency) of words in newspaper articles as mixtures of several different topics, such as politics, economics, and arts, where each has its own preference for certain words over others. These latent variables, corresponding to mixing proportions of topics, characterize a document's statistical properties. Furthermore, these variables are assumed to lie on some nonlinear spaces such that changes among them correspond to smooth-varying high-dimensional data.

Cognitive psychology offers other motivating examples that can help explain human perception and long-term memory. One hypothesis is that prototypes are stored in discrete attractors—that is, human perception and long-term memory should be built upon the framework of continuous attractors.<sup>1</sup> This means that objects stored in the visual and other functional cortices actually lie in one

or more low-dimensional manifolds spanned by a collection of continuous attractors separately. Each intrinsic dimension then corresponds to some interpretable variable. Take facial images, for example. We can regard expression, pose, and illumination as intrinsic variables and dimensions to be memorized that are much less than the number of pixels in an image. Therefore, we can infer that memorizing different categories contained in various manifolds or continuous attractors is one possible way of visual perception.

**Memorizing different categories contained in various manifolds or continuous attractors is one possible way of visual perception.**

### Manifold Learning

The degree of Euclidean locality in a manifold depends on the data distribution and on how we define the neighborhood. If the data are subject to Gaussian distribution and the neighborhood is large enough, the entire Euclidean space can be referred to as a manifold. Therefore, a manifold is a generalization of Euclidean space, but not a special one. Assuming that data reside in a low-dimensional manifold that is then embedded within a high-dimensional space, the goal of manifold learning is to uncover such low-dimensional manifolds from a collection of high-dimensional data points.

According to the low-dimensional space that the manifold-learning

methods attempt to recover, we can roughly divide them into three categories:

- *distance-preserving methods*, such as isometric mapping (Isomap),<sup>2</sup> maximum variance unfolding (MVU),<sup>3</sup> and Hessian locally linear embedding (HLLE);<sup>4</sup>
- *angle-preserving methods*, such as conformal eigenmaps;<sup>5</sup> and
- *proximity-preserving methods*, such as Laplacian eigenmaps (LE)<sup>6</sup> and locally linear embedding (LLE).<sup>7</sup>

To provide a better understanding, we will discuss several representative algorithms in detail.

### Distance Preserving

Isomap claims that the geodesic distance  $g(x_i, x_j)$  between any two points  $x_i$  and  $x_j$ , a shortest path along the manifold, should be preserved when data are mapped into a low-dimensional subspace through some mapping function  $f$ . That is,  $\|f(x_i) - f(x_j)\| = g(x_i, x_j)$ .<sup>8</sup> To achieve this goal, Isomap first approximates the geodesic distances by either computing Euclidean distances when two data points are close or computing graph distances by a well-known shortest-path algorithm (such as the Dijkstra or Floyd algorithms) when the points are at a greater distance.

Then, a classical dimension-reduction method, multidimensional scaling (MDS), is performed to reduce the dimension. More specifically, MDS first uses a centering matrix  $H = I - (1/n)11^T$  to centralize the geodesic distance matrix  $D$ . That is,  $B = -(1/2)HDH$ . Here each element  $d_{ij}$  in the matrix  $D$  is a geodesic distance between different  $x_i$  and  $x_j$ ,  $I$  is the identity matrix, and  $1$  is a column vector of all ones. By performing a spectral decomposition of  $B$ , we have  $B = U\Lambda U^T$ , where  $U$  is an orthogonal matrix, which means each column vector

is orthogonal to other column vectors;  $\Lambda$  is a diagonal matrix, which means each element not on the diagonal will be equal to zero; and  $T$  means matrix transposition. To keep the Gram matrix positive semidefinite, the negative elements in the diagonal of matrix  $\Lambda$  are set to 0. Finally, the data  $X$  are reduced to a low-dimensional subspace using an orthogonal matrix  $U_d$  corresponding to  $\Lambda_d$  (that is,  $Y = XU_d$ ).

Similar to Isomap, MVU tries to preserve the distances between neighboring points when these points are mapped into the low-dimensional subspace:  $\|y_i - y_j\| = \|x_i - x_j\|$ . Meanwhile, MVU tries to ensure that the remaining data points can be pulled as far as possible—that is,  $\max \Sigma_{i,j} \|y_i - y_j\|^2$ . Furthermore, MVU assumes that data in the low-dimensional subspace should be invariant to translation. MVU utilizes semidefinite programming to approximate the goal. Specifically, preserving the distances between neighboring points is converted into a constraint on the Gram matrix  $B_{i,i} + B_{j,j} - 2B_{i,j} = \|x_i - x_j\|^2$ , and  $B$  should be positive semidefinite—that is,  $B$  is positive semidefinite if all its eigenvalues are nonnegative. Here,  $B_{i,j} = \langle y_i, y_j \rangle$  is the inner product of two data points in the low-dimensional space. Pulling distant points as far as possible is modeled by the cost function  $\text{tr}(B)$ , where  $\text{tr}$  means the trace of matrix  $B$ . Moreover, the constraint  $\Sigma_{i,j} B_{i,j} = 0$  restricts data in the low-dimensional subspace to be invariant to translation. Once the Gram matrix  $B$  is learned by a semidefinite programming algorithm, we use the same spectral decomposition procedure as Isomap or classical MDS to achieve the dimension reduction.

### Proximity Preserving

As opposed to the Isomap approach, LE attempts to keep points that are

in close proximity close in the low-dimensional embedding in a local sense. First, LE defines a locally weighted matrix  $W$ , where each element  $w_{i,j}$  of  $W$  is a weighted distance between samples  $x_i$  and  $x_j$ . The distance can be set to one if  $x_i$  and  $x_j$  are connected in a local neighborhood, otherwise it is zero. Furthermore, the distance can also be defined by a Gaussian heat kernel:

$$w_{i,j} = \exp^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}}$$

where  $\sigma$  is a predefined parameter. Then LE minimizes the cost function  $\Sigma_{i,j} (y_i - y_j)^2 w_{i,j}$  subject to the constraint  $y^T D y = 1$ , where  $d_{ii} = \Sigma_j w_{i,j}$ .

**Manifold learning can provide an auxiliary way to enhance the human ability to analyze data.**

The constraint is to guarantee that data will not collapse to one point in the low-dimensional space. Consequently, the cost function can be alternatively solved via the generalized eigenvalue problem  $Ly = \lambda Dy$ , where  $L = D - W$  is the graph Laplacian. Then the low-dimensional data set  $Y$  consists of the  $d$  eigenvectors of the solution corresponding to the bottom  $d$  nonzero eigenvalues.

LE assumes that the linear least-square fit of each point from its neighbors should be preserved when being projected onto a low-dimensional space. It also uses the similar constraints to prevent the selected data from collapsing to one point and to guarantee that the

low-dimensional data representation is invariant to translation.

### Other Methods

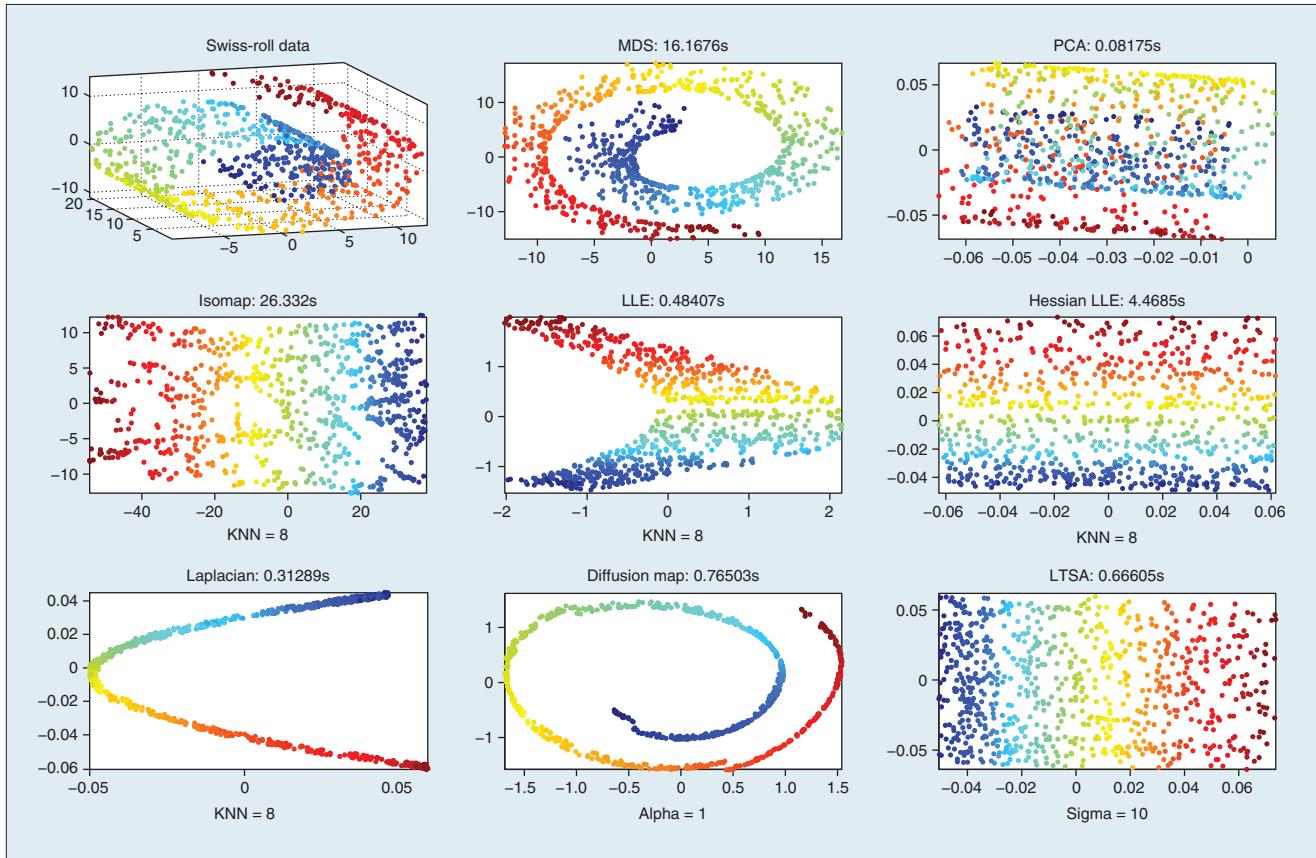
Besides the aforementioned dimension-reduction methods, HLLE maps a collection of local tangent spaces obtained by PCA to a low-dimensional subspace by using a Hessian matrix as the alternative of the graph Laplacian.<sup>4</sup> Similarly, local tangent space alignment (LTSA) aligns a collection of local tangent subspaces (in which each subspace is separately learned from a group of small-scale data using PCA) onto a global coordinate system.<sup>9</sup> Given a random walk on the data, diffusion maps calculate the eigenvectors of the transition matrix and use them to embed the data into a lower space such that the diffusion distances are equal to the Euclidean distances in the embedded space.<sup>10</sup> Conformal eigenmaps project the reduced subspace obtained by LLE or LE into a lower subspace to maximally preserve angles between each data point and its neighboring points.<sup>5</sup>

Although there are various motivations behind the development of manifold-learning algorithms, most share three common steps:<sup>11</sup>

1. neighbor selection,
2. weight computation, and
3. spectral decomposition, or convex optimization under some normalization constraints.

The first step includes defining local data patches for which data points will be connected by non-zero weighted edges in the graph. For example, the weights can be obtained by heat kernel,<sup>6</sup> binary edge,<sup>6</sup> and least square.<sup>7</sup> The second step uses a weighted graph in an attempt to connect data points, either sparsely or densely. Once the graph is built,

# DATA VISUALIZATION



**Figure 2.** The 2D reduced results of 3D Swiss-roll data using eight dimension-reduction methods. Each subplot lists the method and corresponding computation times. The x-axis gives the parameter setting, if necessary. The upper left figure is the input data.

spectral decomposition strategies and optimization techniques can be applied to extract either the first largest  $m$  or the first smallest  $m$  eigenvalues and their corresponding eigenvectors. Also, it is necessary to remove the smallest eigenvalue and its corresponding eigenvector to preserve the translation invariant and the rotation invariant when we apply LLE, HLLE, and Laplacian eigenmap methods as well as diffusion maps. Semidefinite programming has increasingly attracted attention because it provides an alternative of the aforementioned strategies to obtain more reasonable subspaces—for example, conformal eigenmaps<sup>5</sup> and MVU.<sup>3</sup>

To better understand manifold-learning algorithms, we ran six well-known manifold-learning algorithms and two classical dimension-reduction

algorithms to reduce a 3D Swiss-roll data set to a 2D subspace. (The eight dimension reduction algorithms are available at <http://www.math.ucla.edu/~wittman/mani>.) Figure 2 shows that, unlike MDS and PCA, most manifold-learning algorithms successfully unravel the data set in the 2D subspace.

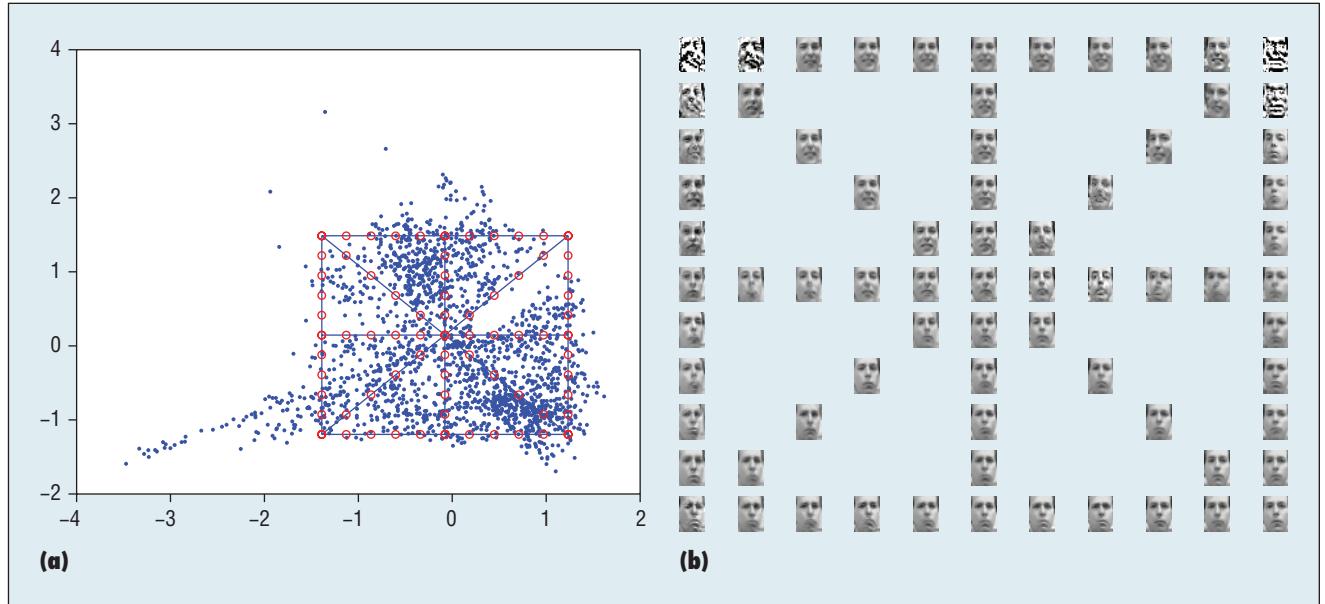
## Applications

Since the introduction of Isomap and LLE techniques, manifold learning has evoked the creation of numerous applications, including data visualization. Humans can only perceive 3D objects, whereas the data we face are higher dimensional. Therefore, manifold learning is a viable way to visualize data from underlying low dimensions. Even in cases where the intrinsic dimension is higher than

three, manifold learning can still provide an auxiliary way to enhance the human ability to analyze data.

Figure 3 illustrates manifold learning's visualization ability. Figure 3b shows that facial expressions vary along the vertical axis and poses vary along the horizontal axis. Therefore, we can infer that the Frey database has at least two intrinsic features (dimensions) including facial expression and pose. The reduced dimension is remarkably smaller than the original one if one pixel is equal to one dimension.

If we assume that each person's images reside on his or her own manifold, then we can recognize different people by calculating the shortest distance from the unknown image to each person's manifold. Motivated by earlier seminal works,<sup>1,2,7</sup>



**Figure 3.** Frey face images ( $20 \times 28$  pixels, 1,956 examples). (a) The results were obtained by first reducing 491 landmark face images to 2D space using LLE with a neighbor factor  $KNN = 8$ . Then all the 1,956 face examples were mapped into the 2D space by a radial-basis-function (RBF) mapping model that has the universal approximation property, using the parameter  $\sigma^2 = 100$ . Next, we randomly sampled two points and used them as the upper-left and lower-right corner points for a rectangle and sampled 11 evenly spaced points along each of the rectangle's boundary and diagonal lines. (b) Then we reconstructed these points with a RBF inverse-mapping model.<sup>12</sup>

researchers have recently proposed numerous algorithms to achieve face recognition, or more broadly, object recognition based on manifold learning.

### Issues and Future Directions

Although manifold learning has attracted more and more attention of researchers from all kinds of fields, three major issues remain: dimension estimation, scalability, and evaluation.

### Dimension Estimation and Scalability

Other researchers have attempted to estimate the intrinsic dimension of manifolds with high accuracy. In Isomap,<sup>2</sup> a turning point that corresponds to an intrinsic dimension can appear on the curve of a reconstruction error versus the reduced dimension. A similar phenomenon has also been observed for conformal eigenmaps.<sup>5</sup> However, it is still an open

problem to determine which intrinsic dimensions will have a crucial influence on manifold learning's effectiveness.

The second issue is how to project samples outside the training set, or out of the sample, into the reduced low-dimensional space. One way is to consider the graph-based manifold-learning methods as instances of kernel PCA (KPCA). If different kernel functions are derived from the methods we have mentioned here, then it is possible to process KPCA using the Nyström method.<sup>13</sup> Therefore, we can easily use low-rank approximation when dealing with out-of-samples; that is, samples can be projected onto low-dimensional subspaces without involving additional computation procedures.

An alternative is to employ landmark-based techniques, where we use a few landmark points to reduce dimensionality and then use the relationship between the landmark points and the out-of-samples to compute

the corresponding low-dimensional descriptions of the out-of-samples.<sup>13</sup> It is worth noting that the landmarks reduce the computational complexity of the given manifold-learning method.

### Evaluation

To evaluate manifold learning's capability to reveal intrinsic dimensions from high-dimensional data, roughly speaking, we can use several criteria:

- the capability to visually unfold data containing an intrinsic flat manifold—that is, whether data lying on a Swiss roll or S-Curve can be mapped onto a 2D flat manifold without severe distortion;
- the capability to avoid a short-circuit phenomenon, which means that the recovery of intrinsic structures suffers from the incorrect connection of two distant data points;
- the robustness of the overall methodology in the presence of noise; and

- for those algorithms that try to **preserve proximity**, such as LLE, whether the neighborhood relationship between each point and its neighboring points can be visually preserved when being projected onto a low-dimensional space (such as the order of nearest neighbors).

(See related work for further discussion on the last point.<sup>14</sup>)

Besides these criteria, we can also evaluate manifold-learning algorithms based on specific applications such as classification. **Although numerous evaluation criteria exist, it is difficult to make a fair comparison between them.** This might be why there are currently so many manifold-learning algorithms being proposed.<sup>8</sup>

## Other Potential Issues

Since manifold learning was first proposed, many theoretical issues have been solved.<sup>14</sup> For example, Zhenyue Zhang and Hongyuan Zha obtained error bounds of LTSA based on the error estimation of local neighborhoods.<sup>9</sup> Xiaoming Huo, Xuelei (Sherry) Ni, and Andrew K. Smith proposed the first performance bound of a manifold-learning algorithm; using the matrix perturbation theory, they obtained the worst-case upper bound of the computed intrinsic subspace and the embedded intrinsic subspace for LTSA.<sup>11</sup> Mira Bernstein and her colleagues claimed that the pairwise distance in the embedding space recovered by Isomap asymptotically approximates the geodesic distances between points on the manifold.<sup>15</sup>

Yair Goldberg and his colleagues also attempted to uncover common shortcomings of manifold learning algorithms.<sup>14</sup> They raised several important points:

- It is uncertain whether the low-dimensional manifolds obtained by

manifold-learning algorithms were a true approximation of the actual intrinsic dimensions.

- It is uncertain whether an algorithm of favorable asymptotical properties exists.
- It is unclear whether manifold-learning algorithms can maintain stability in the presence of noise.

By studying these points, they claimed that several representative manifold-learning algorithms—including LLE, Isomap, HLLE, LTSA, Laplacian Eigenmap, and diffusion map—could be categorized as normalized output algorithms. That is, a low-dimensional space is obtained from high-dimensional data by solving a convex optimization problem under particular normalization constraints, such as a translation invariant and a rotation invariant. **Although it is desirable to perform normalization to preserve manifold structures and invariants, normalization often results in instability when the true underlying manifold has different variances along different directions.** In more extreme cases, when the ratio of maximum variance in the principal axis to other variances in the subordinate axes is greater than two, such a normalization-output algorithm cannot successfully recover the underlying manifold. Furthermore, they also pointed out that an asymptotical convergence does not always hold, especially when there is noise.

Because of space limitations, we were unable to explore additional discussions on manifold learning such as time complexity, embedding types, and implicit assumptions. See related works for more details on these topics.<sup>8,11</sup> A useful **manifold-learning resource** is also available at <http://www.iipl.fudan.edu.cn/~zhangjp/literatures/MLF/INDEX.HTM>.

**B**ecause a manifold is a generalization of Euclidean space and most data are highly nonlinear, manifold learning will certainly become a powerful nonlinear data analysis tool in various domains. However, there is still a long way to go. **Manifold-learning algorithms are problematic if data properties are not properly considered.** There must be more quantitative methods available to evaluate manifold-learning algorithms and better ways to automatically select intrinsic dimensions. In the future, it might be useful to further investigate theoretical directions, such as the existence and uniqueness of low-dimension manifolds and the convergence rate of manifold-learning algorithms, as well as low-computational demands. □

## Acknowledgments

We acknowledge support from National Natural Science Foundation of China (NSFC) grants (numbers 60635030, 60975044, and 60703003), the 973 program (numbers 2010CB327900 and 2006CB705500), and the Shanghai Leading Academic Discipline project number B114. We also acknowledge the invaluable comments and constructive suggestions by Fei Sha of the University of Southern California and our anonymous reviewers.

## References

1. H.S. Seung and D.D. Lee, "The Manifold Ways of Perception," *Science*, vol. 290, no. 5500, 2000, pp. 2268–2269.
2. J.B. Tenenbaum, V.D. Silva, and J.C. Langford, "A Global Geometric Framework for Nonlinear Dimensionality Reduction," *Science*, vol. 290, no. 5500, 2000, pp. 2319–2323.
3. K.Q. Weinberger, F. Sha, and L.K. Saul, "Learning a Kernel Matrix for Nonlinear Dimensionality Reduction," *Proc. 21st Int'l Conf. Machine Learning*, (ICML 2005), ACM Press, 2004, p. 106.
4. D.L. Donoho and C. Grimes, "Hessian Eigenmaps: Locally Linear Embedding Techniques for High-Dimensional

## THE AUTHORS

- Data," *Proc. Nat'l Academy of Sciences*, vol. 100, no. 10, 2003, pp. 5591–5596.
5. F. Sha and L.K. Saul, "Analysis and Extension of Spectral Methods for Nonlinear Dimensionality Reduction," *Proc. 22nd Int'l Conf. Machine Learning* (ICML 2005), ACM Press, 2005, pp. 784–791.
6. P. Niyogi, "Laplacian Eigenmaps for Dimensionality Reduction and Data Representation," *Neural Computation*, vol. 15, no. 6, 2003, pp. 1373–1396.
7. S.T. Roweis and L.K. Saul, "Nonlinear Dimensionality Reduction by Locally Linear Embedding," *Science*, vol. 290, no. 5500, 2000, pp. 2323–2326.
8. L. Cayton, *Algorithms for Manifold Learning*, tech. report, Univ. of Calif., San Diego, 2005.
9. Z. Zhang and H. Zha, "Principal Manifolds and Nonlinear Dimension Reduction via Local Tangent Space Alignment," *SIAM J. Scientific Computing*, vol. 26, no. 1, 2005, pp. 313–338.
10. R.R. Coifman et al., "Geometric Diffusions as a Tool for Harmonic Analysis and Structure Definition of Data: Diffusion Maps," *Proc. Nat'l Academy of Sciences*, vol. 102, no. 21, 2005, pp. 7426–7431.
11. X. Huo, X.S. Ni, and A.K. Smith, "A Survey of Manifold-Based Learning Methods," *Recent Advances in Data Mining of Enterprise Data*, T.W. Liao and E. Triantaphyllou, eds., World Scientific, 2007, pp. 691–745.
12. J. Zhang, S.Z. Li, and J. Wang, "Manifold Learning and Applications in Recognition," *Intelligent Multimedia Processing with Soft Computing*, Y.P. Tan, K.H. Yap, and L. Wang, eds., Springer-Verlag, 2005, pp. 281–300.
13. C.J.C. Burges, "Geometric Methods for Feature Extraction and Dimensional Reduction: A Guided Tour," *Data Mining and Knowledge Discovery Handbook: A Complete Guide*

**Junping Zhang** is an associate professor in the Shanghai Key Laboratory of Intelligent Information Processing and School of Computer Science at Fudan University. His research interests include machine learning, pattern recognition, image processing, biometric authentication, and intelligent transportation systems. Zhang has a PhD in pattern recognition and intelligent system from the Chinese Academy of Sciences. He is a member of IEEE. Contact him at jpzhang@fudan.edu.cn.

**Hua Huang** is an associate professor in the School of Electronics and Information Engineering at Xi'an Jiaotong University. His research interests include image processing and machine learning. Huang has a PhD in communication and information engineering from Xi'an Jiaotong University. He is a member of IEEE. Contact him at huanghua@xjtu.edu.cn.

**Jue Wang** is a professor in the Key Laboratory of Complex Systems and Intelligence Science and the Institute of Automation at the Chinese Academy of Sciences. His research interests include machine learning, data mining, knowledge representation, artificial neural networks, genetic algorithms, and multiagent systems. Wang has a PhD in pattern recognition and intelligent system from the Chinese Academy of Sciences. He is a senior member of IEEE. Contact him at jue.wang@mail.ia.ac.cn.

- for Researchers and Practitioners,  
O.Z. Maimon and L. Rokach, eds.,  
Springer Science and Business, 2005,  
pp. 59–92.
14. Y. Goldberg et al., "Manifold Learning:  
The Price of Normalization," *J. Machine*  
*Learning Research*, vol. 9, Aug. 2008,  
pp. 1909–1939.
15. M. Bernstein et al., *Graph Approximations to Geodesics on Embedded Manifolds*, tech. report, Stanford Univ., 2000.

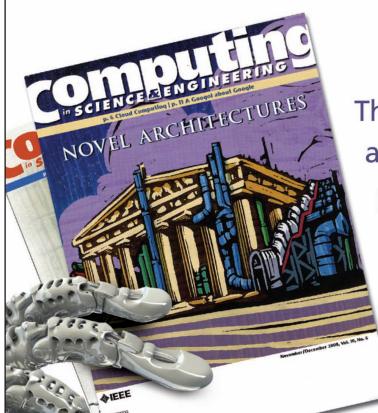
## 2 Free Sample Issues!

A \$26 value

The magazine of computational tools  
and methods for 21st century science.

<http://cise.aip.org>  
[www.computer.org/cise](http://www.computer.org/cise)

Send an e-mail to [jbebee@aip.org](mailto:jbebee@aip.org) to  
receive the two most recent issues of CISE.  
(Please include your mailing address.)



MEMBERS  
**\$47/year**  
for print & online



### Recent Peer-Reviewed Topics:

Cloud Computing  
Computational Astrophysics  
Computational Nanoscience  
Computational Engineering  
Geographical Information Systems  
New Directions  
Petascale Computing  
Reproducible Research  
Software Engineering