Relaxed maximum-variance unfolding

Chenping Hou
Yuanyuan Jiao
Yi Wu
Dongyun Yi
National University of Defense Technology
College of Science
Department of Mathematics and System Science
Changsha 410073, China

Abstract. Nonlinear dimensionality reduction (recently called manifold learning) is crucial in many research fields. Maximum-variance unfolding (MVU) is one of the most important approaches to it. Unfortunately, due to too strict local constraints, MVU cannot unfold the manifold when short-circuit edges appear or the embedded mapping is conformal but not isometric. A relaxed version, relaxed MVU (RMVU), is proposed. Neighbors are adaptively assigned when short-circuit edges appear, and local distance is rescaled when the manifold is assumed to be angle-preserving. RMVU can effectively solve the preceding problems with MVU. More importantly, RMVU also performs better than MVU in general cases, and hence it has huge potential in many fields. Additionally, the proposed two strategies can also be used in other manifold learning algorithms. Experiments, accompanied with numerical comparisons, were performed on both synthetic and real data sets to demonstrate the effectiveness of RMVU. © 2008 Society of Photo-Optical Instrumentation Engineers. [DOI: 10.1117/1.2956373]

Subject terms: nonlinear dimensionality reduction; manifold learning; maximum-variance unfolding.

Paper 071009R received Dec. 27, 2007; revised manuscript received Apr. 3, 2008; accepted for publication May 12, 2008; published online Jul. 17, 2008.

1 Introduction

The problem of nonlinear dimensionality reduction (NLDR) exists in various fields, such as machine learning, pattern recognition, optical image processing, scientific visualization, and multivariable analysis. NLDR focuses on recovering meaningful low-dimensional structures that are hidden in high-dimensional data sets. A simple but prominent example is when a set of pixel images of an individual have been captured from different angles and in different light conditions. The task of NLDR is to identify the two variables, angle and direction of light, given only the high-dimensional pixel data. In many cases of interest, the observed data are assumed to lie on a low-dimensional manifold. The NLDR problem in this form is also known as manifold learning. Given $N$ points $x_1, x_2, \ldots, x_N$ in a high-dimensional subspace of $R^D$, the goal is to find a suitable mapping:

$$f: R^D \rightarrow R^d, \quad y_i = f(x_i) \quad \text{for} \quad i = 1, 2, \ldots, N,$$

(1)

Where $y_i \in R^d$, $i = 1, 2, \cdots, N$, and $d$ is the dimensionality of the embedding space. Due to the large complexity of approximating a function in high-dimensional space, there is no requirement to express the mapping explicitly in applications. One only needs to find the low-dimensional embeddings.

Recently, several new approaches have been developed to address this problem, such as isometric mapping (Isomap), isometric mapping (Isomap), $^{1,2}$ and its modification, C-Isomap, locally linear embedding (LLE); $^{3,4}$ Laplacian eigenmaps, $^{5}$ local-tangent-space alignment (LTSA), $^{6}$ conformal component analysis (CCA), $^{7}$ and maximum-variance unfolding (MVU). $^{8,9,10,11,12}$ Among all of them, MVU is the latest but, most widely used one. It has been applied in sensor localization, $^{13}$ correspondence problems, $^{14}$ subjective mapping for robot localization, $^{15}$ and so on. However, MVU has some unavoidable disadvantages in applications: (1) The computational complexity of MVU is very high; (2) the local constraint of MVU, (viz., that distances between nearby points are assumed to be preserved) is too strict. Weinberger et al. have employed kernel factorization to reduce the computational complexity of the original MVU. $^{16}$ However, as far as we know, no results have been proposed on MVU to deal with the second problem. In particular, (1) when short circuit edges appear, MVU cannot unfold the manifold because one cannot avoid nonuniform wraps and folds; (2) when the mapping $f$ is conformal but not isometric, MVU fails to derive faithful embeddings because of the violation of the local distance-preservation assumption.

The main idea of this paper is to relax the too strict local constraint in two aspects. First, we adaptively construct neighborhood graphs by detecting and eliminating short-circuit edges. Second, local distance is rescaled when the mapping $f$ is conformal. By utilizing these two strategies, traditional MVU is relaxed. This procedure not only extends the application scope of traditional MVU, but also improves the performance of MVU in general cases. Additionally, it is straightforward to embed the two strategies in other manifold learning algorithms. We call this improved MVU relaxed MVU (RMVU).

The rest of this paper is organized as follows. In Sec. 2, MVU is introduced in detail, and the performance of MVU is analyzed. RMVU with two relaxation approaches, together with some preliminary results, is proposed in Sec. 3. In Sec. 4, the application potential of the algorithm is dem-
Hou et al.: Relaxed maximum-variance unfolding

Onstressed on both synthetic and real data sets. The conclusion and further research directions for RMVU are discussed in Sec. 5.

2 Maximum-Variance Unfolding

MVU is a newly proposed manifold learning method that inherits advantages of classical spectral methods and kernel methods. In this section, we first introduce this approach and describe every stage in detail. Performance evaluations of MVU in two special cases are then provided.

2.1 Description of the method

The basic idea of MVU is to maximize the variance of embeddings. It assumes that distances and angles between nearby points are preserved. Low-dimensional coefficients can be considered as local linear transformations of the input data. Obviously, this mapping is isometric.

There are three main stages of MVU. These steps are first listed and then explained in detail.

MVU Algorithm.

Step 1. Construct a neighborhood graph: Define \( \eta_{ij} = 1 \) if and only if points \( i \) and \( j \) have an edge connecting them; otherwise, \( \eta_{ij} = 0 \).

Step 2. Solve the following semidefinite program: Employ the kernel matrix \( K \) with transformation \( K_{ij} = \langle y_i, y_j \rangle \), and maximize trace(\( K \)) with

\[
\begin{align*}
1. & \quad \eta_{ij}(K_{ii} - 2K_{ij} + K_{jj}) = \eta_{ij}\|x_i - x_j\|^2, \\
2. & \quad \sum_{ij} K_{ij} = 0, \\
3. & \quad K \geq 0.
\end{align*}
\]

(2)

Step 3. Derive low-dimensional representations: Assign the \( d \) most significant eigenvectors (corresponding to the \( d \) largest eigenvalues) of the optimal kernel matrix \( K \) as embeddings.

In the first step, there are two commonly used methods to identify neighbors for each point. One simple but efficient way is to examine a fixed number \( k \) of nearest neighbors per point. The other way is to choose all points within an \( \epsilon \)-ball as its neighbors. There are some criteria to determine the number of neighbors per point:

1. The graph formed by linking each point to its neighbors must be connected. Otherwise, MVU will be incorrectly stopped due to the infeasibility of the semidefinite problem shown in Eq. (2).
2. As in LLE, the number of neighbors must be strictly larger than the dimensionality of embedding space. Some margin between \( d \) and \( k \) is generally necessary to obtain a topology-preserving embedding.
3. The size of neighbors cannot be too large; otherwise, the basic assumption of locality is violated.

After constructing a suitable graph, neighborhood indicators \( \eta_{ij} \) are defined to make the following formulas easy to express and explain.

In the second step, the optimization problem in Eq. (2) is deduced as follows. The constraints, which preserve local distance, can be formulated as

\[
\eta_{ij}\|y_i - y_j\|^2 = \eta_{ij}\|x_i - x_j\|^2 \quad \forall i, j.
\]

(3)

To keep the translation invariance of the embedding, the outputs are constrained to be centered, i.e. \( \Sigma_{y_i} = 0 \). MVU attempts to unfold the manifold by maximizing the variance between any two points, i.e., \( \Sigma_{i,j}\|y_i - y_j\|^2 \). Thus, low-dimensional embeddings can be discovered by solving the following optimization problem:

maximize \( \sum_{i,j} \|y_i - y_j\|^2 \) with

\[
\begin{align*}
1. & \quad \eta_{ij}\|y_i - y_j\|^2 = \eta_{ij}\|x_i - x_j\|^2, \\
2. & \quad \sum_{i} y_i = 0.
\end{align*}
\]

(4)

By the simple transformation \( K_{ij} = \langle y_i, y_j \rangle \), the problem in Eq. (4) can be simplified to the semidefinite problem in Eq. (2). It is particularly easy to solve the latter with a facility by such as CSDP, a library for semidefinite problems.

In the final step, it is straightforward to implement the same procedure as in kernel principal-component analysis (KPCA). Embeddings can be determined by spectral factorization of the optimal kernel matrix.

2.2 Performance Analysis of MVU

As mentioned MVU has been widely used in many areas. However, due to its intrinsic defects, it will fail and completely misrepresent the true geometry in some special cases. It will also fail in some real applications. The local constraint of MVU is essentially too strict in practice. We would like to analyze MVU in two important cases and then relax it from the perspective of applications.

It is obvious that smart choice of neighborhood size permits high performance of MVU. The size of \( k \) should not be too large or too small. However, it is hard to have a perfect neighborhood size, especially when the data are contaminated by noise. Short-circuit edges cannot be avoided when one needs to guarantee the connectivity of the neighborhood graph. This can be observed in the following examples.

The data points shown in Fig. 1(a) were uniformly sampled from a noisy two-dimensional spiral. There are altogether 150 points. They were contaminated by noise of normal distribution with zero mean. Their standard variances are 2% of the relevant numerical ranges. It is justifiable to take generating coordinates as one-dimensional coefficients. We have displayed the ideal embeddings in Fig. 1(d). To guarantee the connectivity of neighborhood graphs, experiments show that \( k \) must be equal to or greater
Original data, ideal embeddings, and representations derived by MVU of a spiral: (a) noisy spiral, (b) neighborhood graph, (c) another neighborhood graph, (d) ideal embedding, (e) MVU embedding, (f) another MVU embedding.

Fig. 1 Original data, ideal embeddings, and representations derived by MVU of fishbowl data: (a) original fishbowl data, (b) ideal embedding, (c) MVU embedding.

Fig. 2 Original data, ideal embeddings, neighborhood graphs, and representations derived by MVU of fishbowl data: (a) original fishbowl data, (b) ideal embedding, (c) MVU embedding.

the rim and bottom are overlapped with each other. Embeddings derived by MVU [Fig. 2(c)] are far different from ideal embeddings [Fig. 2(b)]. Therefore, it is necessary to identify an observable invariant of the conformal map, rescale local distances, and relax MVU.

In real applications, it has also been pointed out that MVU fails because of this intrinsic defect. In fact, most manifold learning approaches would fail in these two special cases. Therefore, our strategies can also be used to improve their performance. We only take MVU as an example in this paper.

3 Relaxed Maximum-Variance Unfolding

RMVU is a combination of two strategies in dealing with problems mentioned in the preceding two special situations. In real applications, these two situations are common. Moreover, since the too strict local constraint is relaxed, RMVU usually performs better than MVU in general cases. RMVU integrates two strategies: adaptive neighborhood selection and local distance rescaling. We first introduce RMVU and then explain these two strategies in detail.

3.1 Sketch of RMVU

There are altogether four stages to RMVU.

RMVU algorithm.

Step 1. Selecting neighbors adaptively:

a. Find the nearest \( k \) neighbors of each point and then construct the neighborhood graph.

b. Detect and eliminate short-circuit edges. Determine the neighborhood indicator \( n_{ij} \).

Step 2. Rescaling local distance: Rescale the distance between connected vertices connected by short-circuit edges were required to be preserved in MVU. Therefore, it is unattainable to unfold the manifold and achieve satisfactory results. Detecting and eliminating these short-circuit edges is vital for relaxing MVU and improving its performance.

Another special case is the conformal embedding problem. When \( f \) is not isometric, it is unrealistic to preserve local distances. An example is also given to evaluate the performance of MVU on this benchmark data set.

Five hundred fishbowl data, shown in Fig. 2(a), were generated as follows. 500 points are randomly uniformly selected in a circular disk and then projected stereographically (hence conformally mapped) onto a sphere. Figure 2(b) shows their ideal embeddings. Figure 2(c) shows their representations discovered by MVU in two-dimensional spaces.

Note the high concentration of points near the rim. It is impossible to preserve local distance when we derive true embeddings of a curved fishbowl. Apparently, points near

\[ K_{ij} = \frac{1}{s(i)s(j)} \left( y_i - y_j \right) \]

maximize trace(\( K \)) with

1. \[ \eta_i \left( K_{ii} - 2K_{ij} + K_{jj} \right) = \eta_i \left\| y_i - y_j \right\|^2 \]

2. \[ \sum_{ij} K_{ij} = 0, \]

3. \[ K \succeq 0, \] (5)

Step 4. Deriving low-dimensional representations: Assign the \( d \) most significant eigenvectors (corresponding to the \( d \) largest eigenvalues) of the optimal kernel matrix \( K \) as embeddings.

Note that step 2 and branch b in step 1 are two different techniques to deal with two different kinds of drawbacks of MVU.
3.2 Selecting Neighbors Adaptively

As shown in Figs. 1(b) and 1(c), a short-circuit edge is one that crosses a nonmanifold area, i.e., it is far from the data manifold and connects two points that are intrinsically disconnected. The first step of MVU is vulnerable to short-circuit error if the neighborhood is too large with respect to folds in the manifolds. Even a single short-circuit error can lead to a drastically different (and incorrect) low-dimensional embedding.

Compared with common edges, a short-circuit edge is characterized in that it passes through an area with relatively low sample density. That is, it is far from the data manifold. On the other hand, a common edge is located within the data manifold and cannot cross low-density areas as a short-circuit edge does. It is justifiable to distinguish them according to this characteristic. We first define the so-called deviation \( d(v(i,j)) \) of an edge \( v(i,j) \) in the neighborhood graph of the data set \( \{x_1, x_2, \ldots, x_N\} \) and then explain how to partition these two kinds of edges.

**Definition 1.** Assume \( G \) is a connected graph of the data set \( \{x_1, x_2, \ldots, x_N\} \), and \( v(i,j) \) is the unique edge connecting vertices \( i \) and \( j \). Denote by \( \text{NE}(x_i) = \{x_1, x_2, \ldots, x_k\} \) the set of \( k \) nearest neighbors of \( x_i \), and let \( x_{ij}^{[\text{mid}]} = (x_i + x_j)/2 \). The deviation \( d(v(i,j)) \) of an edge \( v(i,j) \) in neighborhood graph of the data set \( \{x_1, x_2, \ldots, x_N\} \) is defined as follows:

\[
d(v(i,j)) = \frac{1}{|\text{NE}(x_i) \cup \text{NE}(x_j)|} \sum_{x_r \in \text{NE}(x_i) \cup \text{NE}(x_j)} \|x_j - x_{ij}^{[\text{mid}]}\|^2,
\]

where \( |\text{NE}(x_i) \cup \text{NE}(x_j)| \) is the size of \( \text{NE}(x_i) \cup \text{NE}(x_j) \).

This definition is similar to that of the traditional density. Therefore, it can play the same role as density. Naturally, it is a reasonable index to measure the data density in the edge area. The lower the density in this area is, the larger the deviation. It is an index to distinguish edges. When an edge causes a short circuit in a neighborhood graph, its center point is far from data manifold and this deviation tends to be large. On the other hand, edges that are within the scope of the data manifold have small deviations. Therefore, it is reasonable to apply this value to distinguish short-circuit edges from common edges.

There is another critical issue when one implements this technique: How to determine the threshold \( T \) is crucial to the validity of this strategy. Fortunately, the previously defined deviation has the property that it expresses the largest difference between short circuit edges and common edges. Thus, \( T \) can be easily determined based on this property. We sort deviations of all edges and find the largest one among all decrements between items of the first half list. The threshold is the deviation that achieves the most significant difference.

Consequently, we establish the neighborhood selection method as follows.

**Adaptive neighbor selection algorithm.**

Step 1. Construct a neighborhood graph of the \( k \) nearest points for sufficiently large \( k \).

Step 2. Compute the deviation of every edge by applying Eq. \( (6) \).

Step 3. Determine the threshold \( T \). Denote by \( d_{r_i} \) the deviation of the \( i \)th edge in the neighborhood graph. Assume that there are altogether \( P \) edges in this graph.

- a. Sort all \( d_{r_i} \)’s from large to small; denote \( d_{r_1} \geq d_{r_2} \geq \cdots \geq d_{r_{[p/2]}} \).
- b. Assign \( T = d_{r_v} \), where \( v = \arg \max (d_{r_i} - d_{r_{[p/2]}}) \).

Eliminate edges whose deviations are larger than \( T \).

In general, this technique is simple but effective. It works mainly because short-circuit edges traverse areas with lower data densities. In Sec. 4 we describe experiments to demonstrate its potential in applications.

3.3 Rescaling Local Distance

It has also been pointed out that MVU could not solve manifold learning problems when the mapping is conformal but not isometric. This is mainly caused by the violation of the assumption of local distance preservation.

To solve this problem, it is necessary to identify a geometric invariant of conformal maps. Since conformal maps are locally isometric up to a scale factor \( s(y) \), it is straightforward to try to estimate \( s(y) \) at each point of the observed data. In this way, one can restore the original metric structure.

As mentioned in Ref. 3, it is reasonable to choose \( M(x_i) \) as the scalar corresponding to point \( x_i \) if the mapping is conformal and the sampling is uniform. Here \( M(x_i) \) is the average distance of \( x_i \) to its \( k \) nearest neighbors. It was indicated in Ref. 3 that \( (M(x_i)M(x_j))^{1/2} \) is an asymptotically accurate approximation of the conformal scaling factor in the neighborhood of \( x_i \) and \( x_j \).

Note that the previous conclusion holds only when the original sampling is uniform. This may be a severe restriction. However, it can be treated as a necessary trade-off in dealing with large classes of maps. Moreover, in practice, RMU is known to be robust to moderate violations of this assumption. The effect of a violation will be alleviated if the sample size is large enough.

Obviously, if the mapping is conformal, rescaled distances can represent intrinsic structures more faithfully than the original distances. Take the fishbowl data shown in Fig. 2(a) as an example. Compared with the distance between bottom points, the distance between points near the rim of the fishbowl is enlarged. Therefore, it is now possible for MVU to unfold this previously unmanageable example.

In some ideal cases, several preliminary results can be used to verify the appropriateness of this distance rescaling. First, we use a vertical section of the fishbowl to evaluate the performance of this rescaled local distance.
The last inequality holds because $\|x^N_3-x^N_2\|/\|x^N_1-x^N_2\| < 1$, $\|x^F_3-x^F_2\|/\|x^F_1-x^F_2\|$ both tend approximately to one. Therefore, the left-hand side of Eq. (7) is close to one. That is also true for the distance ratio between ideal embeddings, $\|y^N_3-y^N_2\|/\|y^N_1-y^N_2\|$. However, the right side of Eq. (7) is much larger than one if $m_1 \ll m_2$.

2. This kind of rescaling procedure enlarges distances between points that are near the rim and compresses distances between points that are near the bottom.

We can also draw another conclusion from a different perspective.

**Theorem 2.** Assume that $y_1, y_2, \ldots, y_N$ are uniformly sampled. As shown in Fig. 3(b), $y_1, y_2, y_N$ are three nearby points located on a circle with radius $r_Y$, and $\|y^N_1-y^N_2\| = \|y^N_2-y^N_3\| = \delta$. Similarly, $y^F_1, y^F_2, y^F_N$ are located on a circle with radius $r_F$, and we have $r_F \gg r_Y$, $\|y^F_1-y^F_2\| = \|y^F_2-y^F_3\| = \delta$. Let $x^N_1, x^N_2, x^N_3, x^F_1, x^F_2, x^F_3$ be their conformal mapping images. Let $D$ be the diameter of the ball. Assume that $k = 1$, and the nearest points are selected among examples that might have the same coefficients in the vertical directions ($z$ axes). We can conclude that the rescaled distance ratio between nearby and distant points is equal to 1 and much smaller than that of the original, that is,

$$1 = \frac{\|y^N_1-y^N_2\|^2}{\|y^N_1-y^N_3\|^2} = \frac{\|y^F_1-y^F_2\|^2}{\|y^F_1-y^F_3\|^2} = (s_Y)^2, \quad \frac{\|y^F_1-x^F_2\|^2}{\|y^F_2-x^F_3\|^2} = (s_F)^2$$

and

$$s_N = \frac{D^2}{D^2 + (r_N)^2}, \quad s_F = \frac{D^2}{D^2 + (r_F)^2}.$$
Thus, we get the result.

There are some useful facts about this theorem:

1. If the nearby points are selected only within samples of the same height, distances between these nearby points are all rescaled to one.
2. This kind of neighborhood assignment approach cannot guarantee connectivity. However, in view of the results in Theorem 1, it is reasonable to take the rescaled distances as a faithful representation of the original data.

4 Experiments

There are altogether seven experiments described in this section. Two synthetic data sets that have been shown in Fig. 1 and Fig. 2 are employed to evaluate the performance of different approaches. Two real data sets are employed to demonstrate the potential of RMVU in these two special conditions. Another two experiments are provided to illustrate the performance in general cases. The final experiment shows that the performance of Isomap is also improved when we apply these strategies.

Let us first introduce a measure called the mean standard error (MSE).

4.1 Mean Standard Error

In order to measure the divergence between two embeddings, we introduce the MSE. It depends mainly on the shape of manifolds and is invariant under translation, scaling, and symmetrical deformation.

Assume \( Y^{(1)} = [y^{(1)}_1, y^{(1)}_2, \ldots, y^{(1)}_N] \), \( Y^{(2)} = [y^{(2)}_1, y^{(2)}_2, \ldots, y^{(2)}_N] \), where \( y^{(i)}_j \in \mathbb{R}^d \) for \( i=1,2,\ldots,N \) and \( j=1,2 \). First, the two data sets are centered separately, i.e.,

\[
y^{(i)}_j \leftarrow y^{(i)}_j - \text{mean}(Y^{(i)}), \quad \text{where mean}(Y^{(i)}) \text{ denotes the column mean of } Y^{(i)}.\]

Then, every dimension of \( y^{(i)}_j \) is rescaled:

\[
y^{(i)}_j \leftarrow (Y^{(i)}_j - (\text{max}(Y^{(i)}) - \text{min}(Y^{(i)))))/\text{norm}(Y^{(i)}),
\]

where \( \text{max}(Y^{(i)}) \) is a column vector containing the maximum value of every row of \( Y^{(i)} \), and \( \text{min}(Y^{(i)}) \) consists of the minimum value. Here \( \text{norm}(Y^{(i)}) = \text{norm}(Y^{(i)}_1) \) denotes the \( i \)'th element of the difference; \( (Y^{(i)}_j) \) is the \( i \)'th row of \( Y^{(i)} \). Finally, the MSE between \( Y^{(1)} \) and \( Y^{(2)} \) is defined as follows:

\[
\text{MSE}(Y^{(1)}, Y^{(2)}) = \frac{1}{N} \sum_{i=1}^{d} \min[\text{norm}((Y^{(1)}_i - (Y^{(2)}_i), \text{norm}((Y^{(1)}_i - (Y^{(2)}_i)))]
\]

where \( (Y^{(1)}_i) \) represents the \( i \)'th row of \( Y^{(1)} \) and similarly for \( (Y^{(2)}_i) \), and where \( \text{norm}(-) \) represents the 2-norm of a vector.
It is obvious that the MSE is a reasonable measure of the divergence between embeddings.

### 4.2 Spiral

The two-dimensional spiral that is shown in Fig. 1(a) is a benchmark data set to evaluate the performance of different manifold learning methods. We applied an adaptive neighborhood selection approach to eliminate short-circuit edges. Figure 4(a) and 4(b) are the neighborhood graphs corresponding to Fig. 1(b) and 1(c), respectively. Low-dimensional embeddings derived by RMVU (adaptively selecting neighbors) are shown in Fig. 4(c) and 4(d). Comparing with the embeddings discovered by employing MVU [see Fig. 1(e) and 1(f)], it is obvious that RMVU outperforms MVU and represents the original data more accurately.

We also make some numerical comparisons. Since all global manifold learning approaches (for example, Isomap) encounter the short-circuit problem, we compare RMVU with local methods. Since LLE performs best among all local methods, we take it as an example. With different $N$ and $k$, we calculate MSEs between ideal embeddings and those derived by LLE, MVU, and RMVU, respectively. Ten experiments have been done for the same $N$ and $k$ with different noises. We computed their mean and standard deviation (SD). All these results are shown in Table 1.

It is impressive that with different $N$ and $k$, RMVU al-

### Table 1 MSEs and SDs between ideal embeddings for the spiral and those derived by LLE, MVU, and RMVU with different $N$ and $k.$

<table>
<thead>
<tr>
<th>$N$</th>
<th>$k$</th>
<th>LLE Mean</th>
<th>LLE SD</th>
<th>MVU Mean</th>
<th>MVU SD</th>
<th>RMVU Mean</th>
<th>RMVU SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>150</td>
<td>4</td>
<td>1.479e−2</td>
<td>3.312e−3</td>
<td>1.465e−2</td>
<td>6.403e−4</td>
<td>2.047e−4</td>
<td>1.262e−5</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.910e−2</td>
<td>1.606e−3</td>
<td>2.337e−2</td>
<td>6.746e−3</td>
<td>2.154e−4</td>
<td>1.874e−5</td>
</tr>
<tr>
<td>200</td>
<td>4</td>
<td>1.185e−2</td>
<td>3.097e−3</td>
<td>1.250e−2</td>
<td>1.220e−4</td>
<td>1.820e−4</td>
<td>1.417e−5</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.872e−2</td>
<td>3.909e−3</td>
<td>1.938e−2</td>
<td>2.175e−3</td>
<td>2.273e−4</td>
<td>2.935e−5</td>
</tr>
<tr>
<td>250</td>
<td>4</td>
<td>0.959e−2</td>
<td>1.309e−3</td>
<td>1.008e−2</td>
<td>1.789e−4</td>
<td>1.689e−4</td>
<td>3.040e−5</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.725e−2</td>
<td>0.917e−3</td>
<td>1.134e−2</td>
<td>5.459e−4</td>
<td>2.261e−4</td>
<td>1.876e−5</td>
</tr>
<tr>
<td>300</td>
<td>4</td>
<td>0.844e−2</td>
<td>0.324e−3</td>
<td>0.904e−2</td>
<td>8.944e−5</td>
<td>1.415e−4</td>
<td>0.995e−5</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.415e−2</td>
<td>0.153e−3</td>
<td>0.973e−2</td>
<td>2.733e−4</td>
<td>1.709e−4</td>
<td>2.832e−5</td>
</tr>
</tbody>
</table>

It is impressive that with different $N$ and $k$, RMVU al-

![Fig. 5](https://example.com/fig5.png) Low-dimensional coordinates discovered by LLE, MVU, CCA, C-Isomap, and RMVU for fishbowl data with different heights: for height=2, (a) LLE embedding, (b) MVU embedding, (c) CCA embedding, (d) C-Isomap embedding, (e) RMVU embedding; for height=1, (f) LLE embedding, (g) MVU embedding, (h) CCA embedding, (i) C-Isomap embedding, (j) RMVU embedding.
ways yields much smaller means and SDs of the MSE than do (by a factor of about 100) MVU and LLE. RMVU performs the best. This is mainly due to the elimination of short-circuit edges by RMVU. Additionally, RMVU is stable and yields the smallest SDs in most situations. Since LLE is sensitive to noise, it has the largest SDs among the three methods in most cases. In some special cases (such as $N=150, k=5$), MVU may yield a little larger SD since there are different kinds of short-circuit edges.

### 4.3 Fishbowl Data

To evaluate the efficiency of local rescaling, we compared the performance of RMVU (only rescaling local distances), MVU, CCA, C-Isomap, and LLE on fishbowl data. This data set is particularly designed to demonstrate the validity of LLE and C-Isomap. The original data and ideal embeddings were shown in Fig. 2(a) and 2(b) respectively.

We performed experiments on fishbowl data with different heights (i.e., values of the diameter $D$). Two of them are shown in Fig. 5. The five embeddings shown in the first row were derived by LLE [Fig. 5(a)], MVU [Fig. 5(b)], CCA [Fig. 5(c)], C-Isomap [Fig. 5(d)], and RMVU [Fig. 5(e)] on fishbowl data with height=1. Another five representations [Fig. 5(f)–5(j)] derived by the same techniques with height=2 as seen in Fig. 2(a) are shown in the bottom row.

It is obvious that LLE yields the best representation among the five when height=2. However, it fails when the fishbowl is compressed, i.e., height=1. That is mainly caused by the high curvature at some special points on the bowl. MVU fails in both situations because of the too strict local constraint. CCA, which is designed to preserve local angles, also fails when height=1. C-Isomap, which applies the same rescaling procedure before executing Isomap, has comparable performance to RMVU when height=2. However, there is large distortion in its embedding when height=1. Since RMVU rescales local distance and relaxes MVU, it achieves satisfactory and stable results on both kinds of data sets.

Some numerical comparisons are also shown in Table 2. We have computed MSEs between ideal results and embeddings derived by the five methods described. The data sets have different numbers of points and various sizes of neighborhoods. MVU yields the biggest MSE in all situations, and RMVU performs the best when height=1. RMVU yields smaller MSEs than CCA even when the CCA is successful (i.e., height=2). C-Isomap performs about as well as RMVU when height=2 and achieves a little larger MSEs than RMVU when the manifold is compressed. Note that, compared to the situation where height=2, all methods, except LLE, obtain smaller MSEs when height=1. This is because of the difference between local and global approaches. Furthermore, with the increase of $N$, the MSEs of RMVU decrease more rapidly than those of other methods. It is reasonable to say that RMVU is comparable to LLE on large data sets when height=2.

### 4.4 “Lena” Image Data

We applied MVU and RMVU to a data set containing many different “Lena” images. One can generate such samples by translating the “Lena” image across a larger background of random noise. The noises were independent from one ex-

<table>
<thead>
<tr>
<th>$N$</th>
<th>Height</th>
<th>LLE</th>
<th>MVU</th>
<th>CCA</th>
<th>C-Isomap</th>
<th>RMVU</th>
</tr>
</thead>
<tbody>
<tr>
<td>300</td>
<td>1</td>
<td>0.0249</td>
<td>0.0274</td>
<td>0.0255</td>
<td>0.0235</td>
<td>0.0229</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.0226</td>
<td>0.0277</td>
<td>0.0261</td>
<td>0.0229</td>
<td>0.0255</td>
</tr>
<tr>
<td>400</td>
<td>1</td>
<td>0.0218</td>
<td>0.0237</td>
<td>0.0212</td>
<td>0.0193</td>
<td>0.0131</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.0162</td>
<td>0.0239</td>
<td>0.0231</td>
<td>0.0196</td>
<td>0.0212</td>
</tr>
<tr>
<td>500</td>
<td>1</td>
<td>0.0197</td>
<td>0.0213</td>
<td>0.0206</td>
<td>0.0174</td>
<td>0.0108</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.0162</td>
<td>0.0215</td>
<td>0.0209</td>
<td>0.0177</td>
<td>0.0178</td>
</tr>
</tbody>
</table>
Fig. 7 Two-dimensional coordinates discovered by MVU and RMVU from Teapot data with different sample intervals: (a) MVU embedding with interval 4, (b) RMVU embedding with interval 4, (c) MVU embedding with interval 5, (d) RMVU embedding with interval 5.

Fig. 8 Two-dimensional coordinates derived by (a) MVU and (b) RMVU from USPS data.
ample to another. The input consists of \( N = 88 \) grayscale images, with each image containing a \( 32 \times 32 \) “Lena” image superimposed on a \( 61 \times 61 \) background of noise. The first 30 and the last 30 were generated by translating “Lena” by 1 pixel in the vertical direction along the left and right edges, respectively. The rest were generated by translating “Lena” horizontally along the bottom edge. Note that the intrinsic manifold of this data is a line. There would be short-circuit edges connecting points near the corners (left bottom or right bottom). We expect the distances between nearest ideal points to be all equal. Thus, it is reasonable to rescale distances between nearby points.

Embeddings derived by MVU and RMVU are shown in Fig. 6(a) and 6(b). Four real images and their corresponding coordinates are also shown. As expected, since RMVU can eliminate short-circuit edges and rescale local distances, it can derive more realistic embeddings than MVU and represent the intrinsic structure more accurately. Most embeddings lie on a line, and there is much less turbulence.

### 4.5 Teapot Data

The teapot data\(^{17}\) were specially designed to evaluate the performance of MVU. The original data set consists of \( N = 400 \) high-resolution color images showing a porcelain teapot viewed from different angles in the plane. The teapot was viewed under a full 360 deg of rotation. Each image contains \( 76 \times 101 \times 3 \) RGB pixels. Thus, \( D = 76 \times 101 \times 3 = 23,028 \). The expected embedding is in a circle, reflecting the cyclic degree of freedom.

We applied MVU and RMVU to two subsets of these data. The first consists of data that were uniformly sampled with interval 4; the second, with interval 5. Low-dimensional embeddings derived by MVU [Fig. 7(a) and 7(c)] and RMVU [Fig. 7(b) and 7(d)] from these data sets are shown in Fig. 7. Representations of the first are shown in the first row. Some real images, which were accompanied by their embeddings marked by circles, are also shown.

In contrast with the embedding derived by RMVU, coordinates discovered by MVU have some turbulence. Additionally, it seems that the lower the sample density is, the larger the turbulence. The reason is that if the sample density is low, there will be some unnecessary short-circuit edges assigned by searching the nearest \( k \) points. Meanwhile, distances, and thus angles, between nearby points may have a little larger turbulence when a small number of points are selected. Therefore, the embeddings will be more accurate when the local distance is rescaled.

### 4.6 USPS Data

We also compared the performance of MVU and RMVU in a general case, i.e., where there is no prior information about the structure of the data. The USPS data\(^{18}\) consist of 11,000 digit images in ten classes (1100 for each digit). We considered a subset containing \( N = 200 \) handwritten digits (“1” and “2”), consisting of the first 100 examples of each class. The gray images of handwritten numerals are at \( 16 \times 16 \) resolution and converted to \( D = 256 \)-dimensional vectors.

We have applied MVU and RMVU to these real data. The embeddings, accompanied by real images, are shown in Fig. 8. Since the strict local constraint of MVU is relaxed, it is obvious that RMVU achieves better results than MVU. Most of the digits are well clustered in the embedding of RMVU. In contrast, the representations of MVU have large overlaps between the two classes. It is much easier to separate embeddings of RMVU than those of MVU. Thus, RMVU also performs better than MVU even in general cases.

### 4.7 Breast Cancer Data

Since LLE achieves satisfactory results in most applications, we compared RMVU with LLE and MVU on the Diagnostic Wisconsin Breast Cancer data set.\(^{19}\) It consists of 569 instances, with 32 attributes of each case. This data set can be divided into two classes: malignant and benign. Excluding the first attribute (ID) and the second attribute (diagnosis), we chose the first 98 instances, which contained 64 malignant cases and 34 benign cases for illustration. There is no prior structure information about these data either.

We show the two-dimensional embeddings of LLE [Fig. 9(a)], MVU [Fig. 9(b)] and RMVU [Fig. 9(c)]. Circles represent the malignant instances, and dots represent the benign ones.

As seen from Fig. 9, embeddings of LLE and MVU cannot represent the class information of the original data in a proper way. The two classes are heavily overlapped with each other. Fortunately, RMVU achieves satisfactory results. Representations derived by RMVU are approxi-
4.8 Evaluation of Improved Isomap

Another advantage of RMVU is that two approaches in RMVU can also be utilized by other manifold learning algorithms. We have applied these two strategies in Isomap. Experiment shows that the performance of Isomap is improved. This implies that our strategies can be used in more areas.

We employed Isomap, Improved Isomap (i.e., Isomap with adaptive neighborhood selection), and rescaled local distance on spiral data. Figure 10(a) and 10(b) show the neighborhood graphs of Isomap and Improved Isomap. Corresponding results are shown in Fig. 10(c) and 10(d), respectively. For visualization, the x axes in (c) and (d) were the generating coordinates. Thus, the ideal embedding is a straight line.

As expected, since the short-circuit edges are eliminated and local distance is rescaled, Improved Isomap could derive more practical embeddings than traditional Isomap. As seen from Fig. 10(c), Isomap fails in this situation. The nearly straight-line appearance of Fig. 10(d) implies that Improved Isomap is of high application potential.

In a word, RMVU performs better than MVU in most cases. Furthermore, two strategies in RMVU can be used directly in other manifold learning approaches. Thus, our improvement is of high application potential.

There are also some deficits of RMVU: (1) The two innovations are intuitive; there is only little theoretical justification for them; (2) as with MVU, the computational complexity of RMVU is high. One useful approach to solve the latter problem is graph kernel factorization proposed by Weinberger et al. in Ref. 10.

5 Conclusions

In this paper, we have modified MVU to its relaxed version, RMVU. We adaptively assign neighbors when traditional approaches fail and rescale local distance when the mapping is conformal. RMVU not only extends the application scope of traditional MVU, but also improves its performance in general cases. It is more practical to employ RMVU on data even with no prior structure information. Additionally, the two approaches can be directly applied in other manifold learning approaches.

Six databases, including artificially synthesized data and real data, have been used to evaluate the performance of RMVU. Experimental results show that the proposed method can derive more practical embeddings in most common cases and improve the performance of the original MVU.

In the future, we will continue our work to make RMVU fast through kernel factorization and other approaches. How to apply the two strategies in other manifold learning algorithms will also be part of our future work.

Acknowledgment

We would like to thank two anonymous reviewers for their constructive suggestions. We also thank Kilian Q. Weinberger, the creator of the USPS data sets, and the creators of the Diagnostic Wisconsin Breast Cancer Database for providing us with useful data. Thanks are due to the National Basic Research Program of China under grant No. 2005CB321800, the National Natural Science Foundation of China under grant No. 60673090, and the Graduate School of National University of Defense Technology (B070201) for their support.

References


Chenping Hou received his BS degree in applied mathematics from the National University of Defense Technology, Changsha, China, in 2000. He is now a PhD candidate in the Department of Mathematics and System Science. His current research interests include dimensionality reduction, manifold learning, and computer vision.

Yuanyuan Jiao is a PhD candidate in the Department of Mathematics and System Science at the National University of Defense Technology in Changsha, China.

Yi Wu is a professor in the Department of Mathematics and System Science at the National University of Defense Technology in Changsha, China. He earned his bachelor's and master's degrees in applied mathematics at the National University of Defense Technology in 1981 and 1988. He worked as a visiting researcher at New York State University in 1999. His research interests include applied mathematics, statistics, and data processing.

Dongyun Yi is a professor in the Department of Mathematics and System Science at the National University of Defense Technology in Changsha, China. He is now working as a visiting researcher at the University of Warwick.