Local Orthogonality Preserving Alignment for Nonlinear Dimensionality Reduction

Tong Lin*, Yao Liu, Bo Wang, Liwei Wang, Hongbin Zha
The Key Laboratory of Machine Perception (Ministry of Education), Peking University
Beijing, China
{lintong,liuyao}@pku.edu.cn, wangbo1204@gmail.com, {wanglw,zha}@cis.pku.edu.cn

Abstract

We present a new manifold learning algorithm called Local Orthogonality Preserving Alignment (LOPA). Our algorithm is inspired by the Local Tangent Space Alignment (LTSA) method that aims to align multiple local neighborhoods into a global coordinate system using affine transformations. However, LTSA often fails to preserve original geometric quantities such as distances and angles. Although an iterative alignment procedure for preserving orthogonality was suggested by the authors of LTSA, neither the corresponding initialization nor the experiments were given. Procrustes Subspaces Alignment (PSA) implements the orthogonality preserving idea by estimating each rotation transformation separately with simulated annealing. However, the optimization in PSA is complicated and multiple separated local rotations may produce globally contradictive results. To address these difficulties, we first use the pseudo-inverse trick of LTSA to represent each local orthogonal transformations with the unified global coordinates. Then the orthogonality constraints are relaxed to be an instance of semidefinite programming (SDP). Finally a two-step iterative procedure is employed to further reduce the errors in orthogonal constraints. Extensive experiments show that LOPA can faithfully preserve distances, angles, inner products, and neighborhoods of the original data sets. In comparison, the embedding performance of LOPA is better than PSA and comparable to state-of-the-art algorithms like MVU and MVE, while the complexity of LOPA is significantly lower than other competing methods.

1. Introduction

Manifold learning is a large class of nonlinear dimensionality reduction methods operated in an unsupervised manner, with each method attempting to preserve a particular geometric quantity such as distances, angles, proximity, or local patches. Since the two pioneering work published on *Scinece* in 2000, Isomap [19] and LLE [14, 15], manifold learning [16] has been a significant topic in data visualization and pattern classification. Today the huge amount of data from imaging devices, bioinformatics, and financial applications are usually high-dimensional, thus there is an imperative need to overcome the "curse of dimensionality" [3]. A direct solution is the dimensionality reduction approach that transforms the high-dimensional data into a low-dimensional embedding space. However, traditional methods like PCA or MDS fail to discover nonlinear or curved structures of the input data. In contrast, manifold learning methods are suitable for unfolding the nonlinear structures into a flat low dimensional embedding space. Therefore, these methods have found a wide variety of applications, for instance, microarray gene expression, 3D body pose recovery, face recognition and facial expression transferring. See [11] for some recent applications based on manifold alignment.

According to the methodology in [26], existing manifold learning methods can be roughly divided into three categories: (1) distance-preserving methods, including Isomap [19], MVU [22, 23], MVE [18], and RML [9]; (2) angle-preserving methods, e.g. conformal eigenmaps [17]; and (3) proximity-preserving methods, such as LLE [14, 15], HLLE [4], Laplacian eigenmaps (LE) [1], LTSA [27], and NPPE [13], which align local weights or neighborhood for each data point into a global coordinates space. Due to recent advancement, here we point out that there exists the fourth category: (4) patch-preserving methods, such as LMDS [25], and MLE [21], which align each linear patch of moderate size with other patches in order to construct the global representation. In addition, several special methods occurred to be seemingly excluded by the four main categories, such as manifold sculpting [5] and NeRV [20].

Most previous manifold learning methods focus on one particular perspective in order to preserve a single geometric quantity. In this way, for instance, a proximity-preserving method often performs poorly if viewed from other perspectives such as maintaining distances and angles.

[6] addressed the following basic question: how do we define a *faithful* embedding that preserves the local structure of neighborhoods on the manifold? In other words, can we find some fundamental clues to handle distances, angles, and neighborhoods in a comprehensive way? Their answer is the *Procrustes measure*, which computes the distance between two configurations of points after one of the configuration is rotated and translated to best match the other. As the translation vector can be omitted by centering each point set, the computation of Procrustes measure boils down to finding the best rotation (orthogonal) matrix. Then they proposed two algorithms, greedy Procrustes (GP) and Procrustes subspaces alignment (PSA), to minimize the suggested measure. GP is a progressive method that relies on the selection of a basis point and the embeddings produced by GP may not maintain the global structure of the input data (e.g. the cylinder data of Fig. 3 in [6]). On the other hand, PSA performs the global embedding by finding each local orthogonal transformation separately with complicated simulated annealing (SA) and then aligning multiple local PCA subspaces together. However, there is a risk that these local orthogonal transformations may produce an incompatible global embedding since each orthogonal transformation is estimated separately.

We agree with [6] that the Procrustes measure is one reasonable clue to be preserved in manifold learning. To circumvent the difficulties in PSA, in this paper we propose a new algorithm called Local Orthogonality Preserving Alignment (LOPA). Comparison results on synthetic and real data sets demonstrate the good performance of our algorithm.

The rest of the paper is organized as follows. We first discuss some criteria in manifold learning and describe our models in Section 2. Section 3 is devoted to numerically solve the proposed optimization problem, and experimental results are presented in Section 4.

2. Criteria and Models

Generally there are two ways to handle multiple geometric quantities in a comprehensive manner for manifold learning. The first one is to preserve the *Riemannian metrics*, a fundamental notion in Riemannian geometry [2], that determine inner products on tangent spaces at every point. The work by [12] was the first attempt to use Riemannian metrics as a criterion in manifold learning. They provided an algorithm to augment the output of any embedding methods with Riemannian metrics estimated by the Laplace-Beltrami operator; however, they did not develop any new manifold learning algorithm to preserve Riemannian metrics. Here we present a simple model to directly preserve inner products in each neighborhood, and show the inherent difficulties in its optimization.

Given a data set $X = [x_1, \ldots, x_N] \in \mathcal{R}^{m\times N}$ with each data point x_i is a m-dimensional column vector, and the goal of dimensionality reduction is to transform X to $Y = [y_1, \ldots, y_N] \in \mathcal{R}^{d\times N}$ ($d << m$). For each data point x_i, we denote $X_i = [x_{i_1}, \ldots, x_{i_k}]$ as its k nearest neighbors (including itself by setting $x_{i_1} = x_i$), and the neighborhood indices are represented as $\Omega_i = [i_1, \ldots, i_k]$. A direct model to preserve inner products in each neighborhood can be formulated as the following problem to find the optimal Y:

$$\sum_{i=1}^{N} \sum_{\substack{j,l \in \Omega_i,\\ j,l \neq i}} \big(\langle y_j - y_i, y_l - y_i\rangle - \langle x_j - x_i, x_l - x_i\rangle\big)^2. \quad (1)$$

A similar formula occurred in MVU [22, 23], but an equivalent formulation of local isometry, i.e. preserving pairwise distances, is used in the final MVU implementation. We show that the minimization of (1) leads to a standard least squares (LS) problem:

$$\begin{aligned}(1) &= \sum_{i=1}^{N} \|H_k^T S_i^T Y^T Y S_i H_k - H_k^T S_i^T X^T X S_i H_k\|_F^2 \\ &= \sum_{i=1}^{N} \|Q_i^T Z Q_i - W_i\|_F^2 = \sum_{i=1}^{N} \|A_i z - w_i\|^2 \\ &= \|Az - w\|^2,\end{aligned}$$

where $H_k \doteq I - ee^T/k$ is a centering matrix of size k-by-k, I (or I_k) is an identity matrix (of size k-by-k), e is a column vector of all ones (in a proper dimension), S_i is a 0-1 selection matrix for X_i, $Q_i \doteq S_i H_k$, $Z \doteq Y^T Y$, $W_i \doteq H_k^T S_i^T X^T X S_i H_k$ can be computed before hand, $z \doteq vec(Z)$ and $w_i \doteq vec(W_i)$ with the operator $vec(A)$ stacking the columns of A into a long column vector, and A and w are formed by stacking all A_i and w_i together. Note that we use the well-known equality $vec(AXB) = (B^T \otimes A)vec(X)$ in the step obtaining z from Z, where $\otimes$ denotes the Kronecker product.

However, this LS problem is rank deficient in solving a N^2-dimensional vector z with only Nk^2 equations, thus having an infinite number of solutions in most cases such that $k^2 << N$. As rank$(Y) = d$ for common cases when $d < N$, the result Y obtained by eigen-decomposition of $Z = Y^T Y$ is usually a poor embedding. One remedy is to explicitly incorporate the rank constraint of Y into the LS problem. But the fixed rank or low rank LS problem poses great challenges for finding reasonable embeddings for high dimensional data sets. Furthermore, the huge sizes of $A \in \mathcal{R}^{Nk^2\times N^2}$ and $w \in \mathcal{R}^{Nk^2\times 1}$ can be problematic in storage even for a small data set. For instance, the number of matrix entries in A is 64 billions if $N = 1024$ and $k = 8$.

Notice that the complexity of above inner product model (1) essentially comes from the quadratic term $Y^T Y \in$

$\mathcal{R}^{N\times N}$ in the inner product representation. In order to reduce the complexity, we resort to an alternative way based on *local alignments preserving orthogonality* (or isometry). The Local Tangent Space Alignment (LTSA) method [27] provides an elegant framework for neighborhood alignments:

$$\min_{Y,\{L_i\}} \sum_{i=1}^{N} \|YS_iH_k - L_i\Theta_i\|_F^2, \tag{2}$$

where $\Theta_i \in \mathcal{R}^{d\times k}$ is the d-dimensional PCA coordinates for X_i, $L_i \in \mathcal{R}^{d\times d}$ is a local affine transformation, and F denotes the matrix Frobenius norm. The cost function of (2) is then one order about Y (rather than Y^TY in the above inner product model). Then using a pseudo-inverse trick, for fixed Y the optimal affine transformation can be represented as $L_i = YS_iH_k\Theta_i^\dagger$ where $\Theta_i^\dagger$ is the Moore-Penrose generalized inverse of Θ_i. Hence the cost function of (2) can be formulated as $tr(YBY^T)$, where $B \doteq \sum_{i=1}^{N} S_iH_k(I - \Theta_i^\dagger\Theta_i)(I - \Theta_i^\dagger\Theta_i)^TH_k^TS_i^T \in \mathcal{R}^{N\times N}$ (see the derivations in [27]). Finally, by imposing the unit covariance constraint $YY^T = I_d$, the LTSA algorithm obtains the optimal Y given by the eigenvectors corresponding to the d smallest positive eigenvalues of B. However, general linear transformations can *not* preserve local geometric quantities such as distances and angles.

A nature extension is to restrict the linear transformations L_i in the set of orthogonal matrices, leading to our LOPA model:

$$\begin{aligned} \min_{Y,\{L_i\}} \quad & \sum_{i=1}^{N} \|YS_iH_k - L_i\Theta_i\|_F^2, \\ s.t. \quad & L_iL_i^T = I_d, \quad i = 1,\ldots,N. \end{aligned} \tag{3}$$

The LOPA model (3) is similar to PSA, except that PSA directly aligns the low-dimensional embedding with the input data X (without the use of PCA projection). Again using the pseudo-inverse trick to represent L_i, the LOPA model can be rewritten as

$$\begin{aligned} \min_Y \quad & tr(YBY^T), \\ s.t. \quad & YC_iY^T = I_d,\ i = 1,\ldots,N \end{aligned} \tag{4}$$

where $C_i \doteq G_iG_i^T \in \mathcal{R}^{N\times N}$ with $G_i \doteq S_iH_k\Theta_i^\dagger \in \mathcal{R}^{N\times d}$. An earlier work of the ONPP [8] method shares a similar idea:

$$\begin{aligned} \min_Y \quad & tr(YMY^T), \\ s.t. \quad & Y = V^TX, V^TV = I_d, \end{aligned}$$

where $M \in \mathcal{R}^{N\times N}$ is a known matrix, and Y is obtained by an orthogonal transformation of X. However, ONPP has only one orthogonality constraint and is a linear projection.

3. Optimizations

3.1. Orthogonality constraint problems

The LOPA model (4) is a minimization problem with multiple matrix orthogonality constraints. Minimization with orthogonality constraints [24] plays an important role in many applications of science and engineering, such as polynomial optimization, combinatorial optimization, eigenvalue problems, sparse PCA, p-harmonic flows, 1-bit compressive sensing, matrix rank minimization, etc. See [24] for descriptions of some recent applications. Three types of problems are considered in [24]:

$$\min_X \mathcal{F}(X), \quad s.t.\ X^TX = I,$$

$$\min_X \mathcal{F}(X), \quad s.t.\ X^TMX = K,$$

$$\min_{X_1,\ldots,X_q} \mathcal{F}(X_1,\ldots,X_q), s.t. X_i^TM_iX_i = K_i, i = 1,\ldots,q$$

where $\mathcal{F}$ is a known differentiable function, M, M_i, and K_i are given positive definite and nonsingular symmetric matrices. It is generally difficult to solve these problems because the orthogonality constraints can lead to many local minimizers and several type of these problems are NP-hard. No guarantee can be made for obtaining the global minimizer, except for a few simple cases such as finding the extreme eigenvalues.

Generally the approaches to solve orthogonality constraint problems can be roughly classified into two categories [24]: (1) *feasible methods* that strictly satisfy the orthogonality constraints during iterations, including matrix re-orthogonalization and generating trial points along geodesics; (2) and *infeasible methods* that relax the constraints by penalizing their violations and thus generate infeasible intermediate points, such as various penalty, augmented Lagrangian, and SDP relaxation methods.

In this paper the LOPA model (4) is solved by an infeasible method, since the strict orthogonality constraints are rarely satisfied except for a few *intrinsically flat* data set with zero Gaussian curvature everywhere, such as the Swiss role data. Specifically the SDP relaxation method is used to solve the LOPA problem, with details given in the following subsection.

3.2. Relaxation models for LOPA

A most straightforward way to simplify (4) is to replace the multiple constraints with just a single combined constraint $YCY^T = I_d$, where $C = \sum_{i=1}^{N} C_i/N$. This simplification can be derived from the Lagrangian function

$$\mathcal{L}(Y,\{W_i\}) = tr(YBY^T) - \frac{1}{N}\sum_{i=1}^{N} tr\big(W_i(YC_iY^T - I_d)\big),$$

where each W_i is a Lagrangian multiplier matrix. If assuming all the multiplier matrices are identical as W, then the penalization term can be written as

$$tr\big(W(Y(\frac{1}{N}\sum_{i=1}^{N} C_i)Y^T - I_d)\big).$$

Thus we can obtain an overly simplified model:

$$\min_Y \ tr(YBY^T),\ s.t.\ YCY^T = I_d. \tag{5}$$

If considering each dimension of Y, then the optimal Y is simply given by the eigenvectors corresponding to the d smallest positive generalized eigenvalues of $(B, C + \delta I_N)$. Here δI_N is a small regularization term to avoid singularity. However, this overly simplified model is not amenable to embedding curved manifold data, though yielding satisfactory results on intrinsically flat data like Swiss roll.

A more practical way is to replace the difficult orthogonal constraints by easier trace constraints, leading to the following relaxation model:

$$\min_Y tr(YBY^T),\ s.t. tr(YC_iY^T) = d, i = 1, \ldots, N. \tag{6}$$

Compared with the rigid orthogonality constraint $YC_iY^T = I_d$, the trace constraint $tr(YC_iY^T) = d$ at each data point only loosely specifies the sum of the diagonals of YC_iY^T. By setting $K \doteq Y^TY$ and using the trace property $tr(ABC) = tr(BCA) = tr(CAB)$, the model (6) can be rewritten as

$$\begin{aligned} \min_K \quad & tr(BK), \\ s.t. \quad & K \succeq 0, tr(C_iK) = d, i = 1, \ldots, N, \end{aligned} \tag{7}$$

where $K \succeq 0$ shows it is a positive semidefinite matrix with rank d by its definition.

3.3. Connection to MVU

It is interesting to connect the LOPA model (7) with the MVU model [22, 23] given by:

$$\begin{aligned} \max_Y \quad & tr(K), \\ s.t. \quad & K \succeq 0,\ tr(ee^TK) = 0, \\ & K_{ii} - 2K_{ij} + K_{jj} = D_{ij}, j \in \Omega_i, \end{aligned} \tag{8}$$

where $D_{ij} \doteq \|x_i - x_j\|^2$ is the squared distance between two neighbors (x_i and x_j). The last constraint in (8) is just $\|y_i - y_j\|^2 = D_{ij}$ represented by K, showing that the main purpose of MVU is to preserve distances between neighbor points. The second constraint enforces that the embeddings of all data points should be centered on the origin:

$$\begin{aligned} \sum_i y_i = 0 \quad &\Rightarrow \quad \sum_{ij} y_i^T y_j = 0 \Rightarrow Ye = 0 \\ &\Rightarrow \quad tr(e^TY^TYe) = 0 \Rightarrow tr(ee^TK) = 0. \end{aligned}$$

The objective function of MVU is derived as followings:

$$\begin{aligned} tr(K) \quad &= \quad tr(Y^TY) = \sum_i \|y_i\|^2 \\ &= \quad \frac{1}{2N}\sum_{ij}(\|y_i\|^2 + \|y_j\|^2 - 2y_i^Ty_j) \\ &= \quad \frac{1}{2N}\sum_{ij}\|y_i - y_j\|^2, \end{aligned}$$

where the zero mean constraint is used in the third equality. Therefore, it is clear that MVU attempts to unfold the curved manifold by maximizing the averaged squared distance between any two embedding points (need not to be k-nearest neighbors) under the distance preserving constraint, thus getting its algorithmic name.

We can see that the objective function $\max tr(K) = \min -tr(IK)$ of MVU (8) is similar to $\min tr(BK)$ of LOPA (7). However, there are approximately $Nk/2$ constraints of pairwise distances in the MVU model. In contrast, LOPA (7) has only N constraints, thus having lower complexity than MVU.

3.4. Solution to LOPA

It is well known that the LOPA model (7) is a standard formulation of semi-definite programming (SDP) and the optimal K can be solved by any off-the-shelf convex optimization toolbox like *sdpt3*, *csdp*, and *sedumi*. In general the obtained K may not satisfying the rank d constraint coming from the definition $K = Y^TY$. Then by eigen-decomposition of K we get an *initial* solution of the embeddings, $Y_0 = VD^{\frac{1}{2}}$, where D and V are the top d eigenvalue diagonal matrix and the corresponding eigenvectors of K, respectively. Here $D^{\frac{1}{2}}$ denotes the diagonal matrix formed by the square roots of the top d eigenvalues.

Recall that we only solved the relaxation version (7) to approximate the original LOPA problem (4). Starting from the initial SDP solution Y_0, it is usually possible to find better Y such that both the cost function $tr(YBY^T)$ and the penalty terms $\|YC_iY^T - I_d\|_F^2$ can be further decreased. Here we directly use the two-step iterative procedure suggested by the Appendix of [27]:

1. For fixed Y, solve $\min_{L_i} \|YS_iH_k - L_i\Theta_i\|_F^2$ to obtain an optimal orthogonal transformation L_i for each x_i. This is the standard *orthogonal Procrustes problem* (See Algorithm 12.4.1 of [7]), with solution $L_i = U_iV_i^T$ where $YS_iH_k\Theta_i^T = U_i\Sigma_iV_i^T$ is the singular value decomposition (SVD).

2. For the fixed $\{L_i\}, i = 1, \ldots, N$, solve the least squares (LS) problem $min_Y \sum_i \|YS_iH_k - L_i\Theta_i\|_F^2$ to update Y.

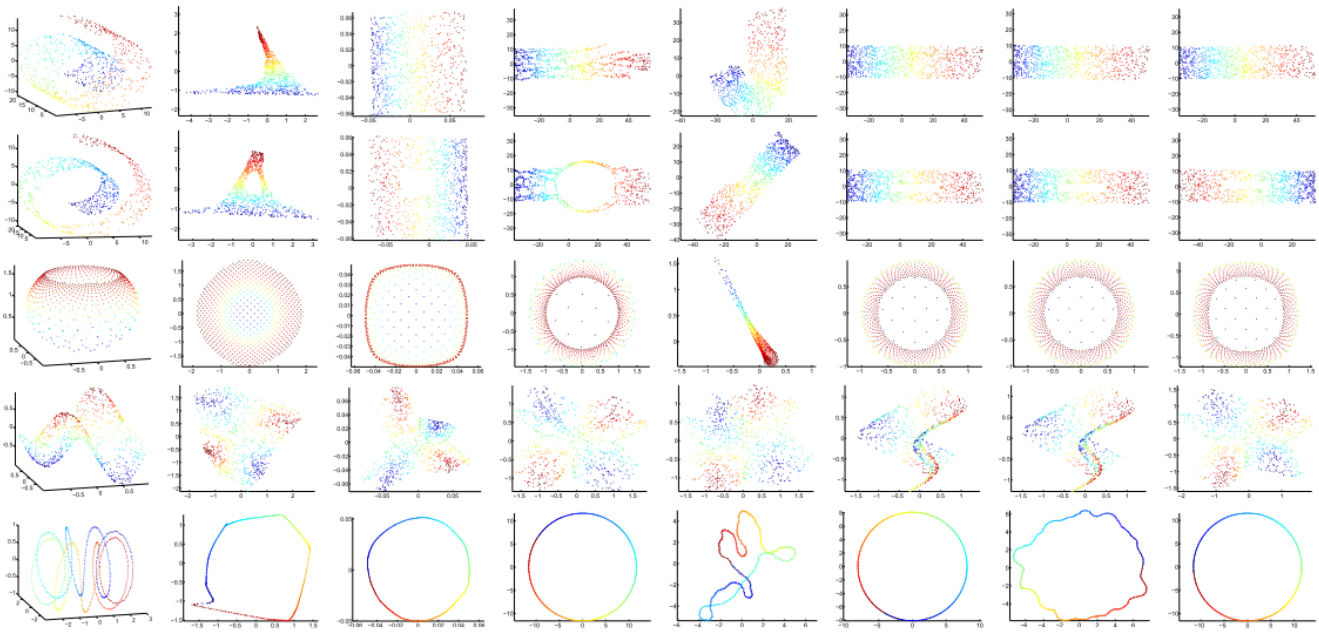

Figure 1. 3D to 2D results on synthetic data. From top to bottom: Swiss roll, Swiss hole, punctured sphere, twin peaks, and toroidal helix.

Table 1. R_{kdist}.

DATASET	σ	LLE	LTSA	ISOMAP	PSA	MVU	MVE	LOPA
SWISS	0	99.53%	100.00%	48.14%	18.13%	4.22%	1.67%	**0.42%**
ROLL	0.03	99.00%	100.00%	44.98%	16.54%	8.98%	6.14%	**1.09%**
SWISS	0	99.45%	100.00%	41.81%	18.42%	3.84%	1.64%	**0.42%**
HOLE	0.03	99.08%	100.00%	38.95%	19.49%	8.77%	10.09%	**0.94%**
PUNCTURED	0	80.91%	99.83%	79.93%	53.59%	44.70%	32.15%	**24.06%**
SPHERE	0.03	80.80%	99.83%	79.87%	57.70%	44.64%	32.26%	**23.72%**
TWIN	0	185.82%	99.83%	32.76%	25.62%	27.53%	34.38%	**9.94%**
PEAKS	0.03	53.81%	99.85%	36.33%	21.67%	26.86%	28.19%	**10.29%**
TOROIDAL	0	97.07%	99.77%	99.62%	14.92%	1.13%	**0.18%**	0.65%
HELIX	0.03	97.07%	99.77%	99.49%	8.74%	1.16%	**0.22%**	0.67%

3. Repeat the above two steps until $\|Y^{(t+1)} - Y^{(t)}\|_F / \|Y^{(t)}\|_F \leq \epsilon$, or $t \geq t_{max}$.

4. Experiments

We compare our algorithm with other dimensionality reduction method, including PCA, LLE, LTSA, Isomap, PSA, MVU, and MVE. Aside from showing results on synthetic data, we also show visualization results on pose varying data and motion sequence. Moreover classification performance is evaluated on low-dimensional embeddings of five diverse data sets.

To produce a quantitative evaluation, we introduce four averaged measures reflecting geometric changes before and after the embedding in each neighborhood. These measures are relative errors in distances, relative errors in angles, relative errors in inner products among any three neighbors, and change rates in k-nearest neighborhood:

$$
\begin{aligned}
R_{kdist} &= \frac{\sum_{i=1}^{N}\sum_{j=2}^{k}|(x_{i_j}-x_i)^2-(y_{i_j}-y_i)^2|}{\sum_{i=1}^{N}\sum_{j=2}^{k}(x_{i_j}-x_i)^2}, \\
R_{kangl} &= \frac{\sum_{i=1}^{N}\sum_{j=3}^{k}|\angle x_{i_j}x_ix_{i_2}-\angle y_{i_j}y_iy_{i_2}|}{\sum_{i=1}^{N}\sum_{j=3}^{k}|\angle x_{i_j}x_ix_{i_2}|}, \\
R_{kinner} &= \frac{\sum_{i=1}^{N}\sum_{j=2}^{k}|<x_{i_j},x_i>-<y_{i_j},y_i>|}{\sum_{i=1}^{N}\sum_{j=2}^{k}|<x_{i_j},x_i>|}, \\
R_{knn} &= \frac{1}{kN}\sum_{i=1}^{N}(k-|\Omega(x_i)\cap\Omega(y_i)|).
\end{aligned}
$$

Table 2. R_{kangl}.

DATASET	σ	LLE	LTSA	ISOMAP	PSA	MVU	MVE	LOPA
SWISS	0	51.27%	34.98%	20.60%	11.23%	0.64%	0.45%	**0.41%**
ROLL	0.03	48.04%	37.86%	21.35%	10.08%	4.28%	4.35%	**4.14%**
SWISS	0	38.83%	35.16%	27.20%	10.18%	0.67%	0.50%	**0.44%**
HOLE	0.03	34.45%	36.70%	31.32%	13.06%	4.75%	4.95%	**4.17%**
PUNCTURED	0	9.74%	43.75%	25.87%	32.27%	23.60%	20.52%	**7.94%**
SPHERE	0.03	9.76%	43.75%	25.99%	45.45%	23.38%	20.67%	**8.94%**
TWIN	0	29.53%	11.88%	21.17%	16.98%	18.39%	8.77%	**6.50%**
PEAKS	0.03	14.41%	14.94%	22.84%	11.98%	15.70%	9.99%	**5.92%**
TOROIDAL	0	4.36%	**0.06%**	4.36%	2.55%	3.49%	4.36%	3.08%
HELIX	0.03	4.33%	**0.24%**	4.33%	1.74%	2.94%	3.74%	3.23%

Table 3. R_{kinner}.

DATASET	σ	LLE	LTSA	ISOMAP	PSA	MVU	MVE	LOPA
SWISS	0	99.53%	100.00%	64.00%	29.26%	4.24%	1.61%	**0.57%**
ROLL	0.03	98.92%	100.00%	55.58%	26.08%	8.79%	7.06%	**2.30%**
SWISS	0	99.46%	100.00%	66.48%	30.01%	3.76%	1.55%	**0.55%**
HOLE	0.03	99.07%	100.00%	62.94%	30.66%	8.36%	9.47%	**2.05%**
PUNCTURED	0	82.83%	99.88%	95.71%	69.04%	89.09%	65.82%	**34.76%**
SPHERE	0.03	82.85%	99.88%	96.05%	54.11%	88.77%	65.72%	**34.73%**
TWIN	0	251.56%	99.83%	50.12%	42.73%	34.67%	34.05%	**16.24%**
PEAKS	0.03	70.63%	99.84%	53.56%	34.83%	32.21%	28.64%	**16.04%**
TOROIDAL	0	97.06%	99.77%	100.05%	15.31%	1.01%	**0.24%**	0.68%
HELIX	0.03	97.06%	99.77%	100.10%	8.78%	0.97%	**0.32%**	0.85%

Table 4. R_{knn}.

DATASET	σ	LLE	LTSA	ISOMAP	PSA	MVU	MVE	LOPA
SWISS	0	53.58%	38.19%	14.56%	12.31%	0.56%	**0.22%**	0.23%
ROLL	0.03	48.45%	39.39%	13.89%	7.48%	1.17%	1.66%	**0.55%**
SWISS	0	36.02%	36.13%	17.98%	9.77%	0.52%	0.20%	**0.13%**
HOLE	0.03	33.53%	37.20%	21.13%	10.16%	1.06%	1.44%	**0.50%**
PUNCTURED	0	**12.55%**	60.08%	34.78%	38.77%	27.92%	34.80%	24.83%
SPHERE	0.03	**12.30%**	60.11%	34.73%	63.80%	28.29%	34.73%	26.27%
TWIN	0	31.94%	**12.41%**	15.91%	32.16%	40.85%	16.67%	32.19%
PEAKS	0.03	**15.16%**	15.48%	16.16%	27.02%	39.17%	36.45%	33.30%
TOROIDAL	0	**0.16%**	87.50%	5.09%	31.33%	0.19%	**0.16%**	84.77%
HELIX	0.03	0.63%	84.06%	0.16%	32.94%	0.29%	**0.17%**	82.86%

4.1. Synthetic data

Although viewed as "toy data" and shown over and over again in the manifold learning literature, synthetic datasets are often self-explanatory to grasp basic properties of each methods. If one algorithm performs poorly on synthetic data, nobody would believe its good embeddings on real-world datasets. Here five synthetic datasets are used to perform dimensionality reduction from 3D to 2D: Swiss roll, Swiss hole, punctured sphere, twin peaks, and toroidal helix. Every dataset has 800 data points, and the number of neighborhood (k) is set as 8. Two situations are considered, without noise or with Gaussian noise. In the latter case we add Gaussian noise $\mathcal{N}(0, c^2\sigma^2)$ on each dimension of the coordinates, where $\sigma = 0.03$ in our experiments and c is an average distance among one neighborhood for each dataset.

Figure 1 shows the visualization results. We can see that LLE and LTSA can not maintain distances and angles due to their proximity-preserving nature. Other five methods including LOPA attempt to preserve distances or isometry, performing poorly on the punctured sphere because unfolding this curved data into flat will greatly violate the distance preserving criterion. In comparison, Isomap yields unsatisfactory or poor results on Swiss roll, Swiss hole, and punctured sphere; PSA fails to unfold

Table 5. Running time (sec.)

DATASET	σ	LLE	LTSA	ISOMAP	PSA	MVU	MVE	LOPA
SWISS	0	0.257	0.444	7.439	1866.953	652.166	2414.328	144.110
ROLL	0.03	0.261	0.449	6.665	1491.382	598.826	2506.421	143.023
SWISS	0	0.257	0.441	6.861	1780.307	566.924	2848.470	152.897
HOLE	0.03	0.253	0.447	6.834	2780.833	610.673	3499.761	142.318
PUNCTURED	0	0.241	0.457	6.830	533.858	160.716	437.917	99.776
SPHERE	0.03	0.242	0.465	9.443	2419.072	175.084	639.375	138.883
TWIN	0	0.258	0.448	6.922	3631.019	112.332	1588.542	110.066
PEAKS	0.03	0.282	0.455	9.341	4482.946	188.189	1887.481	110.929
TOROIDAL	0	0.206	0.412	6.874	2710.461	436.081	1255.400	85.829
HELIX	0.03	0.209	0.397	7.185	4608.337	226.327	1741.251	141.108

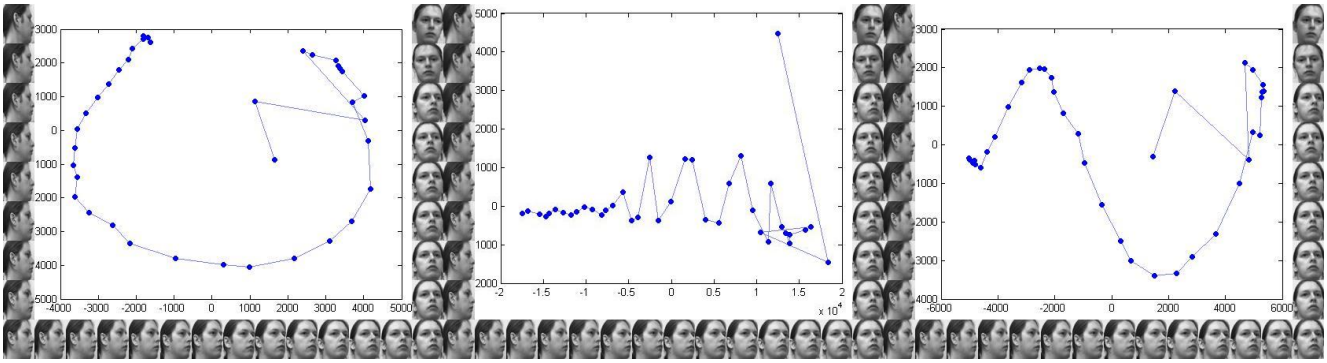

Figure 2. 2D embedding of the UMist face images by using LOPA, MVU and MVE (from left to right, respectively).

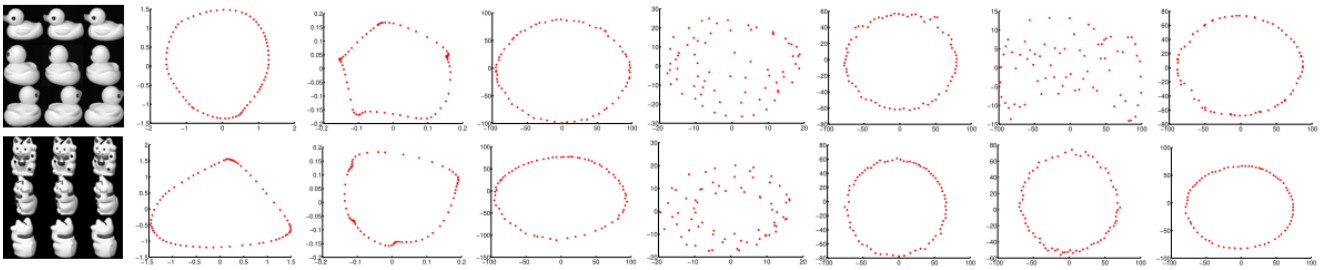

Figure 3. Duck images and cat images in Coil. Note that only part of images are shown in (a).

Swiss roll, punctured sphere, and toroidal helix. The results offered by LOPA, MVU and MVE are very similar except that on twin peaks LOPA performs better.

The results of the four geometric measures on synthetic dataset are listed in Table 1-4, showing that LOPA outperforms other methods significantly in R_{kdist}, R_{kangl}, and R_{kinner}. Table 5 shows the running time on a PC with 2.5GHz CPU and 4G RAM with all algorithms implemented in Matlab. Note that the implementations of LOPA, MVU and MVE use SDPT3 in solving SDP problems for fairness. It is clear that LOPA runs much faster than PSA, MVU, and MVE.

4.2. Real Data: Varying Pose

In this test we compare the ability of recovering the pose transition of facial images and Coil data. Figure 2 compares LOPA with MVU and MVE on 2D embedding of the UMist facial images for one person. Since the inherent structure should be a circular arc, we can see that MVU thoroughly fails to uncover this structure and MVE method yields a

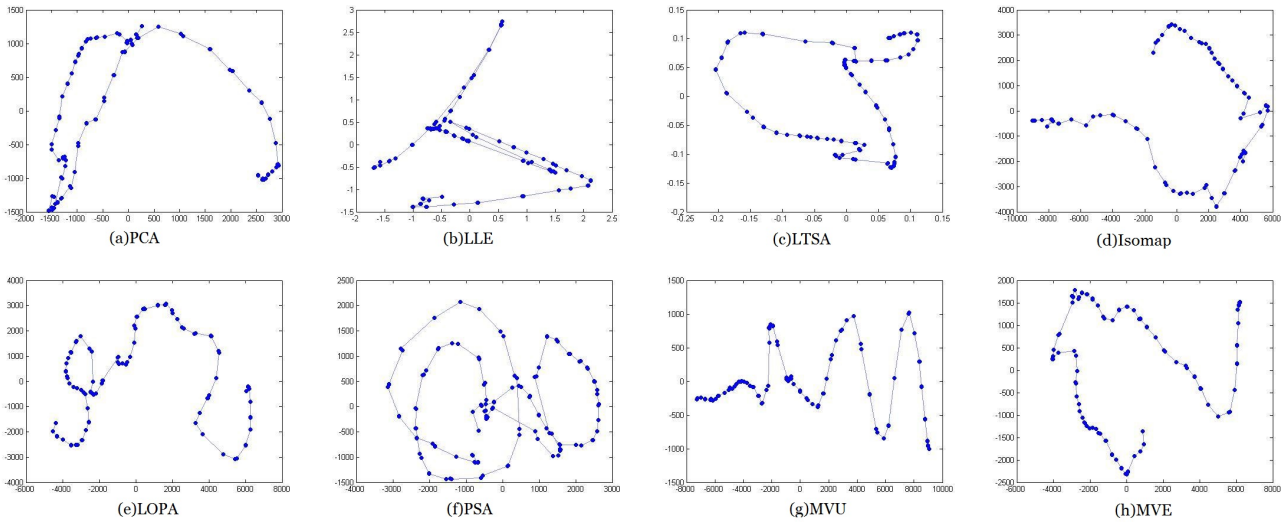

Figure 4. 2D-embedding of images from a basketball video.

Table 6. Runtime and geometric measurements on a basketball video.

measures	PCA	LLE	LTSA	Isomap	PSA	MVU	MVE	LOPA
Time(sec.)	0.359	0.1753	**0.1500**	0.3130	211.7510	9.7360	135.8429	2.1530
$R_k dist$	81.10%	100.00%	100.00%	41.39%	10.61%	49.61%	43.33%	**9.71%**
$R_k angl$	46.74%	56.90%	44.10%	45.14%	37.10%	39.26%	**35.07%**	36.26%
$R_k inner$	82.63%	100.00%	100.00%	60.70%	50.49%	43.94%	56.08%	**42.11%**
$R_k nn$	22.94%	36.06%	22.09%	9.04%	51.21%	9.75%	**7.04%**	7.99%

sin-like curve. In contrast, LOPA reveals the underlying structure as a circular arc. Figure 3 displays 2D embedding results of the "duck" and "cat" images in Coil data. Each group of Coil images, such as the "duck" and the "cat", was captured at every 5 degrees by rotating the objects. Hence the Coil images should have an inherent circle structure. In comparison, LOPA, MVU, and Isomap perform the best on the two groups of Coil images, while PSA fails to detect the circular structure.

4.3. Real Data: Motion Sequence

We use the UCF-sport dataset to explore the low-dimensional representation of a motion image sequence. The 2D embedding of a basketball video clip with 140 frames is shown in figure 4. It can be seen that LTSA, Isomap, LOPA, and MVE can unfold the data into a roughly smooth curve, maintaining the sequential property of the motion. Since the ground-truth low dimensional structure of this dataset is unknown, we also report the quantitative measurements shown in table 6. From these errors we can see that LOPA and MVE perform much better than other algorithms. However LOPA runs much faster than MVE.

4.4. Real Data: Classification

We test the classification performance using k-nearest neighbor classifier after dimensionality reduction. Five data sets are used for this purpose. Both the MNIST dataset and the USPS dataset are handwritten digits. The ORL dataset consist of 400 facial images of 40 persons under different conditions. The HIVA dataset is a drug dicovery dataset with two-class labels. The UCI satellite dataset is an infrared astronomy database with six classes. Some datasets are too large for algorithms like PSA and MVE, so we randomly sampled 600-800 data points from each dataset. Each dataset is preprocessed by using PCA to transform the data into a 100-dimensional space before hand, and then we run different dimensionality reduction algorithms further to embed into a very-low dimension.

Table 7 shows the errors of k-nearest neighbor classifier on embeddings produced by different dimensionality reduction methods. Some parameters are listed in the table. We can see that PCA performs well on most datasets, and takes the first place on two digits data. In comparison, LOPA achieves the best on the HIVA data

Table 7. Test errors of k-nearest-neighbor (KNN) classification (leave-one-out) on low-dimensional data representation produced by dimensionality reduction methods. N: number of data points; D: intrinsic dimension estimated by DrToolbox (also as embedding dimension); K_{dr}: number of neighbors used in dimensional reduction; K_n: number of neighbors used in KNN classifier.

dataset	N	D	K_{dr}	K_n	PCA	LLE	LTSA	Isomap	PSA	MVU	MVE	LOPA
usps	600	10	15	15	**1.67**%	2.83%	2.16%	6.33%	12.33%	5.67%	5.5%	8.83%
mnist	600	20	20	20	**1.34**%	2.67%	45.50%	2.00%	18.33%	2.00%	2.00%	7.00%
orl	400	8	10	3	4.75%	21.00%	40.75%	19.25%	31.75%	6.00%	**4.5**%	5.5%
hiva	800	15	15	3	3.50%	3.75%	**3.25**%	3.63%	3.25%	3.38%	3.37%	**3.25**%
satellite	800	12	15	15	13.63%	17.0%	16.5%	15.00%	23.25&	15.25%	14.75%	**13.62**%

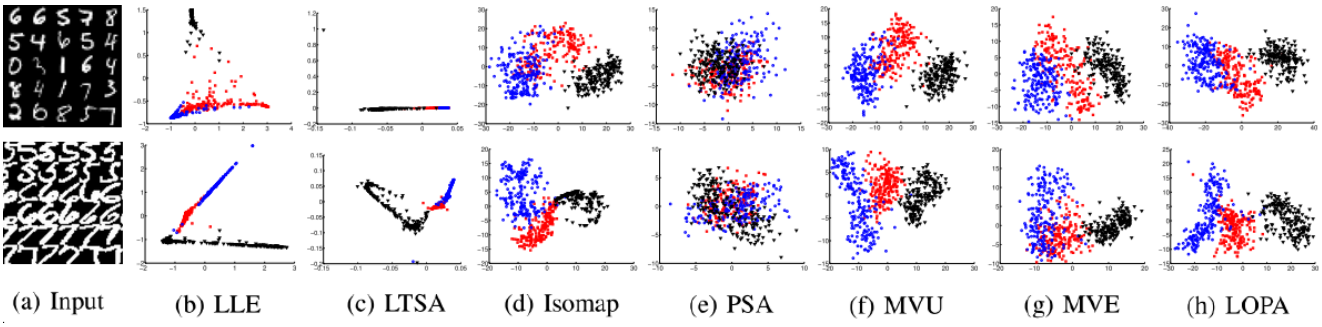

Figure 5. 2D embeddings of selected digits (5-7). Top row: the MNIST data; bottom row: the USPS data. Note only example images are shown in the input (a).

and the UCI satellite dataset. We argue that the main purpose of manifold learning is to faithfully preserve the original geometric properties of the input data when reducing the dimensionality. As class label information being not used, manifold learning may not compete with other discriminant dimensionality reduction methods like Fisher's linear discriminant projection. The survey of [10] has claimed that most of manifold learning methods are even inferior to PCA when using 1-NN classifiers on real datasets.

Figure 5 displays comparison results of 2D embedding on digits 5, 6, and 7 from the MNIST data and from the USPS data. The results indicate that proximity preserving methods like LLE and LTSA often fail to correctly separate the three classes of digits, while PSA yields totally mix-up embeddings on digits. LOPA, together with MVU and MVE, can yield embedding results that are highly separable for different digits. It implies that LOPA can serve as a feature extraction method for digit classification.

5. Conclusion

We proposed a new manifold learning algorithm called Local Orthogonality Preserving Alignment (LOPA). Our algorithm is built upon the neighborhood alignment framework suggested by LTSA, and extend the general linear transformations in LTSA into orthogonal alignments. LOPA overcomes the difficulties in PSA by using the pseudo-inverse trick to avoid multiple incompatible local transformations. Compared with the complicated simulated annealing method used in PSA, we use more efficient SDP relaxation to find the numerical solutions. Experimental results demonstrate that LOPA can produce embedding results comparable to state-of-the-art algorithms like MVU and MVE. Particularly, our method can faithfully preserve distances, angels, inner products, and neighborhood of the input data. On the other hand, the complexity of LOPA is much lower than MVU and MVE because the number of constraints used in LOPA is smaller. Our future work is to investigate efficient numerical methods and to explore some real applications in visualization and classification.

6. Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant Nos. 61375051 and 61075119, the National Basic Research 973 Program of China under Grant No. 2011CB302202, and the Seeding Grant for Medicine and Information Sciences of Peking University under Grant No. 2014-MI-21. T. Lin is the corresponding author.

References

[1] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003. 1

[2] M. P. do Carmo. *Riemannian geometry*. Birkhäuser, Boston, 1992. 2

[3] D. L. Donoho. High-dimensional data analysis: The curses and blessings of dimensionality. *AMS Math Challenges Lecture*, pages 1–32, 2000. 1

[4] D. L. Donoho and C. Grimes. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Sciences*, 100(10):5591–5596, 2003. 1

[5] M. Gashler, D. Ventura, and T. Martinez. Manifold learning by graduated optimization. *IEEE Trans. Systems, Man, and Cybernetics, Part B: Cybernetics*, 41(6):1458–1470, 2011. 1

[6] Y. Goldberg and Y. Ritov. Local procrustes for manifold embedding: a measure of embedding quality and embedding algorithms. *Machine learning*, 77(1):1–25, 2009. 2

[7] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, 1996. 4

[8] E. Kokiopoulou and Y. Saad. Orthogonal neighborhood preserving projections: A projection-based dimensionality reduction technique. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29(12):2143–2156, 2007. 3

[9] T. Lin and H. Zha. Riemannian manifold learning. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 30(5):796–809, 2008. 1

[10] L. Maaten, E. Postma, and J. Herik. Dimensionality reduction: A comparative review. Technical report, TiCC, Tilburg University, 2009. 9

[11] Y. Pei, F. Huang, F. Shi, and H. Zha. Unsupervised image matching based on manifold alignment. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 34(8):1658–1664, 2012. 1

[12] D. Perraul-Joncas and M. Meila. Non-linear dimensionality reduction: Riemannian metric estimation and the problem of geometric discovery. *arXiv preprint arXiv:1305.7255*, 2013. 2

[13] H. Qiao, P. Zhang, D. Wang, and B. Zhang. An explicit nonlinear mapping for manifold learning. *IEEE Trans. Cybernetics*, 43(1):51–63, 2013. 1

[14] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000. 1

[15] L. K. Saul and S. T. Roweis. Think globally, fit locally: unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research*, 4:119–155, 2003. 1

[16] H. S. Seung and D. D. Lee. The manifold ways of perception. *Science*, 290(5500):2268–2269, 2000. 1

[17] F. Sha and L. K. Saul. Analysis and extension of spectral methods for nonlinear dimensionality reduction. In *Proc. Int. Conf. Machine learning*, pages 784–791, 2005. 1

[18] B. Shaw and T. Jebara. Minimum volume embedding. In *Int. Conf. Artificial Intelligence and Statistics*, pages 460–467, 2007. 1

[19] J. B. Tenenbaum, V. De Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000. 1

[20] J. Venna, J. Peltonen, K. Nybo, H. Aidos, and S. Kaski. Information retrieval perspective to nonlinear dimensionality reduction for data visualization. *Journal of Machine Learning Research*, 11:451–490, 2010. 1

[21] R. Wang, S. Shan, X. Chen, J. Chen, and W. Gao. Maximal linear embedding for dimensionality reduction. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 33(9):1776–1792, 2011. 1

[22] K. Q. Weinberger and L. K. Saul. Unsupervised learning of image manifolds by semidefinite programming. *International Journal of Computer Vision*, 70(1):77–90, 2006. 1, 2, 4

[23] K. Q. Weinberger, F. Sha, and L. K. Saul. Learning a kernel matrix for nonlinear dimensionality reduction. In *Proc. Int. Conf. Machine learning*, 2004. 1, 2, 4

[24] Z. Wen and W. Yin. A feasible method for optimization with orthogonality constraints. *Mathematical Programming*, 142(1-2):397–434, 2013. 3

[25] L. Yang. Alignment of overlapping locally scaled patches for multidimensional scaling and dimensionality reduction. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 30(3):438–450, 2008. 1

[26] J. Zhang, H. Huang, and J. Wang. Manifold learning for visualizing and analyzing high-dimensional data. *IEEE Intelligent Systems*, (4):54–61, 2010. 1

[27] Z. Zhang and H. Zha. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *SIAM Journal of Scientific Computing*, 26(1):313–338, 2004. 1, 3, 4