



Curvature-aware manifold learning

Yangyang Li^{a,b,*}

^a Academy of Mathematics and Systems Science Key Lab of MADIS, Chinese Academy of Sciences, Beijing 100190, China

^b University of Chinese Academy of Sciences, Beijing 100049, China

ARTICLE INFO

Article history:

Received 12 June 2017

Revised 9 April 2018

Accepted 6 June 2018

Available online 7 June 2018

Keywords:

Manifold learning

Riemannian curvature

Second fundamental form

Hessian operator

ABSTRACT

One of the fundamental assumptions of traditional manifold learning algorithms is that the embedded manifold is globally or locally isometric to Euclidean space. Under this assumption, these algorithms divided manifold into a set of overlapping local patches which are locally isometric to linear subsets of Euclidean space. Then the learnt manifold would be a flat manifold with zero Riemannian curvature. But in the general cases, manifolds may not have this property. To be more specific, the traditional manifold learning does not consider the curvature information of the embedded manifold. In order to improve the existing algorithms, we propose a curvature-aware manifold learning algorithm called CAML. Without considering the local and global assumptions, we will add the curvature information to the process of manifold learning, and try to find a way to reduce the redundant dimensions of the general manifolds which are not isometric to Euclidean space. The experiments have shown that CAML has its own advantage comparing to other traditional manifold learning algorithms in the sense of the neighborhood preserving ratios (NPR) on synthetic databases and classification accuracies on image set classification.

© 2018 Published by Elsevier Ltd.

1. Introduction

In many machine learning tasks, we often face the problem of how to recognize and how to remove the redundant dimensions of the data points. To solve this problem, a series of methods were proposed. Some algorithms aim to learn the sparse representation of data points [1] and it is a special dimensionality reduction method. At the early stage, the idea and its relative algorithms of using the theory of manifold in dimension reduction were proposed in [2,3], called manifold learning (MAL). In this decade, manifold learning has become a significant research area of machine learning, pattern recognition and image vision, etc. The existing manifold learning algorithms aim to reduce the dimensionality of high dimensional data points, so that the lower dimensional representations could reflect the intrinsic geometrical and topological structure of the high dimensional sample points. In general, the existing MAL algorithms are mainly divided into two classes: global methods and local methods [4]. Global approaches aim to preserve the global geometric structure of the manifold during dimension reduction, such as IsoMap [2] and TCIE [5]; Local approaches attempt to uncover the geometric structures of local patches, such as LLE [3], LEP [6], LPP [7], LTSA [8], and Hessian Eigenmap [9]. The

main purpose of IsoMap is to preserve the geodesic distance between any two high dimensional data points, which can be viewed as a nonlinear extension of Multidimensional Scaling (MDS) [10]. MAL algorithms of local preserving type attempt to inherit and preserve the local geometric structure of the underlying manifold. For instance, LLE preserves the local linear structures of local patches and LEP preserves the local similarities among the data during dimension reduction.

1.1. Manifold assumption

One of the fundamental assumptions of all the existing MAL algorithms is that the input data points are actually samples from a manifold $\mathcal{M}$ which is viewed as a sub-manifold of the ambient feature space. For each algorithm, it requires other special conditions. IsoMap assumes that $\mathcal{M}$ is globally isometric to a convex subset of Euclidean space. TCIE [5] aims to avoid this requirement of IsoMap. However, it still need to assume that $\mathcal{M}$ is isometric to a subset of Euclidean space. Locally preserved manifold learning algorithms [11] visualize the embedded manifold as a collection of overlapping local patches. Each local method preserves a different local geometrical structure of the embedded manifold, so the assumptions for local patches are different. LLE as well as the modified LLE [12] considers that $\mathcal{M}$ is an open sub-manifold and the input data points are dense enough to make the neighborhood of each data point a linear subspace. LEP also regards the neighborhood of each sample as a linear subspace, then set up the corresponding local

* Corresponding author at: Academy of Mathematics and Systems Science Key Lab of MADIS, Chinese Academy of Sciences, Beijing 100190, China.

E-mail address: liyongyang12@mails.ucas.ac.cn

Table 1
Major manifold learning algorithms and their assumptions.

Authors	Year	Algorithm	Manifold assumption
Tenenbaum et al.	2000	IsoMap	Globally isometric to a convex subset of Euclidean space
Roweis et al.	2000	LLE	Locally linear
Belkin et al.	2003	LEP	Locally linear
Donoho et al.	2003	HLLE	Locally isometric to an open, connected subset of Euclidean space
Zhang et al.	2004	LTSA	Locally linear
He et al.	2005	LPP	Linear form of Laplacian eigenmaps
Dollar et al.	2007	LSML	Not locally isometric to Euclidean space
Binbin Lin et al.	2013	PFE	Local isometric to subspace of Euclidean space

weight matrix, where the distance between two adjacent samples is measured by Euclidean metric. HLLE assumes that $\mathcal{M}$ is locally isometric to Euclidean space, so that the null space can be uncovered by the average norm of Hessian matrix of all data points. For LTSA, in each local patch it uses PCA [13] to reduce the dimension of local samples. Therefore, LTSA assumes that each local patch of $\mathcal{M}$ is a linear subspace of Euclidean space. PFE [14] uses parallel vector field to learn a dimension reduction map, where this map induces that $\mathcal{M}$ is locally isometric to Euclidean space. LSML [15] reduces the dimension of $\mathcal{M}$ which is not isometric to Euclidean space. But it regards each local patch of sub-manifold as a linear subspace. All the assumptions of MAL algorithms are shown in Table 1.

As we can see, all the MAL algorithms except LSML require that $\mathcal{M}$ is globally or locally isometric to Euclidean space. In practice, generally speaking, a manifold is far from fitting these assumptions. On the other hand, all the existing MAL algorithms do not reveal information about the reliability and validity of these assumptions. Furthermore, all of them are unable to recognize the difference between non-isometry property and isometry property of a manifold.

1.2. Limitations

By analyzing the well-studied MAL, despite the wide applications of the existing MAL algorithms in many fields, such as: computer vision, pattern recognition, and machine learning, there are still a few limitations and problems remained to be solved.

- **Local linearity assumption:** it requires the input data points to be dense enough to guarantee the local patches being linear subspaces. In practice, there are not enough samples to generate the local patches with small size enough to guarantee the linearity.
- **Parameters sensitivity problem:** the neighbor-size parameter determines the size of local patches. Since the local isometry hypothesis, it requires the neighbor-size small enough. Otherwise, it would break the assumption of the existing MAL algorithms.
- **Locally short circuit problem:** if the embedded manifold is highly curved, the local Euclidean distance between any two points is obviously shorter than the intrinsic geodesic distance.
- **Intrinsic dimension estimation problem:** since local patches are simply taken as tangent spaces, the intrinsic dimension of manifold cannot be determined by the latter accurately, in particular in case of strongly varying curvature.

Since the disadvantages of existing MAL algorithms mentioned above are caused by the assumption that $\mathcal{M}$ is locally or globally isometric to Euclidean space, it is reasonable to eliminate this assumption and to design a new algorithm.

The problem presented in this paper is stated in the following subsection.

1.3. Problem statement

The input data points stated in this paper are denoted as $\{x_1, x_2, \dots, x_N\}$, $x_i \in \mathbb{R}^D$, where N is the total number of data points and D is the original dimension of data points. We assume that these discrete data points lie on a d -dimensional manifold $\mathcal{M}$ embedded in the high dimensional feature space $\mathbb{R}^D$, $d \ll D$, where $\mathcal{M}$ can be viewed as a sub-manifold of $\mathbb{R}^D$. The main purpose of manifold learning is to learn an embedding map f :

$$x_i = f(y_i) + \varepsilon_i, i = 1, \dots, N, \quad (1)$$

where $\{y_1, y_2, \dots, y_N\}$, $y_i \in \mathbb{R}^d$, $d \ll D$, are lower dimensional representations of $\{x_1, \dots, x_N\}$ and $\{\varepsilon_1, \varepsilon_2, \dots, \varepsilon_N\}$ are the corresponding noises. Map f must preserve the geometrical structure of sub-manifold so that the lower dimensional representations can uncover the intrinsic structure of the sub-manifold $\mathcal{M}$.

Under local isometry assumption, the embedding map f must satisfy:

$$\|f(y_i) - f(y_j)\|^2 = \|y_i - y_j\|^2 + o(\|y_i - y_j\|^2), \quad (2)$$

where y_i, y_j are in a same local patch.

For general manifold, the locally isometric condition is not always satisfied, such as sphere [15]. The problem that we aim to solve in this paper is the situation that $\mathcal{M}$ is non-locally isometric to Euclidean space. All the manifold learning algorithms aim to uncover the intrinsic structure of the embedded-manifold $\mathcal{M}$. Thus our method attempts to learn the embedding map f in Eq. (1) under non-isometric condition which is not satisfied by Eq. (2).

2. Geometry background

In this section, we first analyze the local isometry of manifold, then give a geometric interpretation about the local isometry assumption. With this in mind, we uncover the potential limitations of traditional manifold learning algorithms.

2.1. Local isometry

The family of all inner products defined on all tangent spaces of $\mathcal{M}$ is called Riemannian metric g of $\mathcal{M}$. At each tangent space $T_p\mathcal{M}$, the Riemannian metric is a scalar inner product g_p , $p \in \mathcal{M}$. A Riemannian manifold is said to be flat if it is locally isometric [16] to Euclidean space. That is to say, if every point has a neighborhood which is isometric to an open subset of Euclidean space, the Riemannian manifold is called a flat manifold.

Theorem 2.1 [17]. *A Riemannian manifold is flat if and only if its curvature tensor vanishes identically.*

So under the local isometry assumption of traditional MAL, the curvature tensor of sub-manifold $\mathcal{M}$ is zero tensor everywhere. In general, the sub-manifold may be highly curved and not isometric to Euclidean space. In this case, traditional MAL algorithms cannot accurately uncover the intrinsic structure of sub-manifold. Thus, by analyzing the well-studied local isometry, the basic causes of these

limitations for traditional manifold learning algorithms is that the curvature tensor of sub-manifold $\mathcal{M}$ has not been considered.

To our knowledge, there have been several papers seeking to estimate the intrinsic curvature of data points [18–20]. K. I. Kim et al. [18] proposed to add the curvature information to reconstruct the Laplace matrix. This method was only applied on semi-supervised learning as a regularization item and it has been tested in several experiments on four standard datasets. However, the case of unsupervised manifold learning has not been studied and [18] did not give a theoretical analysis to prove the improvement compared with the traditional methods. In [19,20] (Xu et al.), Ricci flow technique was used to rectify the pair-wise non-Euclidean dissimilarities among data points. However, this method treated the curvature information of each edge independently, hence, the relations among edges were neglected. In this paper, we take a more comprehensive approach to formulate our curvature-aware manifold learning algorithm.

2.2. Riemannian sub-manifold

In Riemannian geometry, the geometric structure of sub-manifold $\mathcal{M}$ is determined by two fundamental forms. Riemannian metric g can be viewed as the *first fundamental form* which is used to compute the intrinsic geometric structure of $\mathcal{M}$, such as the geodesic distance, area, volume and so on. The *second fundamental form* (its details will be introduced below) can be used to uncover the extrinsic structure of sub-manifold $\mathcal{M}$ relative to ambient space, such as curvature, torsion and so on. It will help us to see and measure how the sub-manifold $\mathcal{M}$ curves with respect to its ambient space. For Riemannian manifold, the torsion is zero.

2.3. Second fundamental form

Suppose $(\tilde{\mathcal{M}}, \tilde{g})$ is a Riemannian manifold with dimension $\mathcal{D}$ and $(\mathcal{M}, g)$ is embedded in $(\tilde{\mathcal{M}}, \tilde{g})$ with dimension d . At any point $p \in \mathcal{M}$, the ambient tangent space $T_p\tilde{\mathcal{M}}$ divides into two perpendicular linear subspaces $T_p\tilde{\mathcal{M}} = T_p\mathcal{M} \oplus N_p\mathcal{M}$ [17], where $N_p\mathcal{M} \doteq (T_p\mathcal{M})^\perp$ is the normal space and $T_p(\mathcal{M})$ is the tangent space of $\mathcal{M}$ at p . In this paper, we regard Riemannian manifold $\mathcal{M}$ as a Riemannian sub-manifold of $\mathbb{R}^{\mathcal{D}}$. The Riemannian metric g of $\mathcal{M}$ is defined as the induced metric from $\mathbb{R}^{\mathcal{D}}$. Riemannian curvature tensor defined on Riemannian manifold is a 4th order tensor. The curvature operator is represented by the second order derivative on vector field of Riemannian manifold, where the directional derivative is defined as Riemannian connection ∇ . In Riemannian sub-manifold, the Riemannian curvature tensor [21] of sub-manifold is computed with the help of *second fundamental form* denoted as $\mathcal{B}$ (a second order tensor) and is represented by a fourth-order tensor $\mathcal{R}(X, Y, Z, W)$, where X, Y, Z, W are vector fields on $\mathcal{M}$.

In Riemannian sub-manifold, one main task is to compare the Riemannian curvature of $\mathcal{M}$ with that of ambient space $\tilde{\mathcal{M}}$. According to the definition of curvature tensor, we first give the relationship between the Riemannian connection ∇ of $\mathcal{M}$ and $\tilde{\nabla}$ of $\tilde{\mathcal{M}}$ [17]:

$$\tilde{\nabla}_X Y = \nabla_X Y + \mathcal{B}(X, Y), \quad (3)$$

Therefore, the *second fundamental form* measures the difference between the Riemannian connection on $\mathcal{M}$ and the ambient Riemannian connection on $\tilde{\mathcal{M}}$. By the *Guass Equation* [17], the Riemannian curvature $\tilde{\mathcal{R}}(X, Y, Z, W)$ of the ambient space can be decomposed into two components. Since the ambient space we deal with is Euclidean space $\mathbb{R}^{\mathcal{D}}$, so $\tilde{\mathcal{R}}(X, Y, Z, W) = 0$. Then the Riemannian curvature of $\mathcal{M}$ is represented as:

$$\mathcal{R}(X, Y, Z, W) = \langle \mathcal{B}(X, W), \mathcal{B}(Y, Z) \rangle - \langle \mathcal{B}(X, Z), \mathcal{B}(Y, W) \rangle. \quad (4)$$

In order to compute the scalar value of *second fundamental form*, we construct a locally natural orthonormal coordinate frame

$\{\frac{\partial}{\partial x^1}, \dots, \frac{\partial}{\partial x^d}, \frac{\partial}{\partial y^1}, \dots, \frac{\partial}{\partial y^{\mathcal{D}-d}}\}$ of the ambient space $\tilde{\mathcal{M}}$ at point p . The restrictions $\{\frac{\partial}{\partial x^1}, \dots, \frac{\partial}{\partial x^d}\}$ to $\mathcal{M}$ form a local orthonormal frame of $\mathcal{T}_p(\mathcal{M})$. The last $\mathcal{D} - d$ orthonormal coordinates $\{\frac{\partial}{\partial y^1}, \dots, \frac{\partial}{\partial y^{\mathcal{D}-d}}\}$ form a local orthonormal frame of $\mathcal{N}_p(\mathcal{M})$. Under the locally natural orthonormal coordinate frame, the Riemannian curvature of $\mathcal{M}$ in Eq. (4) is represented as:

$$R_{ijkl}^i = \sum_{\alpha} (h_{ik}^{\alpha} h_{jl}^{\alpha} - h_{il}^{\alpha} h_{jk}^{\alpha}). \quad (5)$$

Accordingly, the *second fundamental form* $\mathcal{B}$ under this local coordinate frame is shown as: $\mathcal{B}(\frac{\partial}{\partial x^i}, \frac{\partial}{\partial x^j}) = \sum_{\alpha=1}^{\mathcal{D}-d} h_{ij}^{\alpha} \frac{\partial}{\partial y^{\alpha}}$, with $h_{ij}^{\alpha} (\alpha = 1, \dots, \mathcal{D} - d)$ being the coefficients of $\mathcal{B}(\frac{\partial}{\partial x^i}, \frac{\partial}{\partial x^j})$ with respect to the normal coordinate frame $\{\frac{\partial}{\partial y^1}, \dots, \frac{\partial}{\partial y^{\mathcal{D}-d}}\}$. Accordingly, the embedding map f is redefined as: $f(x^1, x^2, \dots, x^d) = [x^1, x^2, \dots, x^d, f^1, \dots, f^{\mathcal{D}-d}]$, where $x = [x^1, x^2, \dots, x^d]$ are natural parameters. Here $h_{ik}^{\alpha} = \frac{\partial^2 f^{\alpha}}{\partial x^i \partial x^k}$ is the second derivative of the embedding component function f^{α} , which constitutes the Hessian matrix $H^{\alpha} = (\frac{\partial^2 f^{\alpha}}{\partial x^i \partial x^j})$. Hence, to compute the Riemannian curvature of Riemannian sub-manifold $\mathcal{M}$, we just need to estimate the Hessian matrix of the embedding map f . In the next section, we give the estimation of Hessian operator.

3. Curvature-aware manifold learning

In this paper, we just consider the locally geometric structure preserving MAL algorithms, namely LLE, LEP, LTSA and so on. In general, the procedures of this type of algorithms are mainly divided into three steps [22]. The detailed statement is given in the following subsection.

3.1. Manifold learning

In the first step, traditional MAL algorithms partition local patches $\{U_i\}$ on each input point x_i based on the Euclidean metric in ambient space $\mathbb{R}^{\mathcal{D}}$. In general, there are two commonly used methods. One is to choose an ε -ball with x_i as center and then consider the set of all the points in this ball as the neighbors of x_i . The other method is to use K -nearest neighbor method to find the neighbors of each input data point x_i . For these two methods, ε and K are parameters which are highly sensitive to the dimension reduction results of experiments.

In the second step, traditional MAL algorithms aim to construct a weight matrix $\tilde{W}_i$ in each local patch U_i to represent the local geometric structure of sub-manifold $\mathcal{M}$. Different MAL algorithms produce different weight matrices.

The third step is to reconstruct a set of lower dimensional representations $Y = [y_1, \dots, y_N]$, where $y_i \in \mathbb{R}^d$ corresponds to x_i . Here Y is obtained by minimizing a reconstruction error function Φ under some normalization constraints [22].

$$\Phi(Y) = \sum_{i=1}^N \phi(Y_i) = \sum_{i=1}^N \|\tilde{W}_i Y_i\|_F^2, \quad (6)$$

with the normalization constraints $Y^T Y = I, Y^T 1 = 0$ for LLE, LTSA, HLE, and $Y^T \tilde{D} Y = I, Y^T \tilde{D} 1 = 0$ for LEP where $\tilde{D}$ is a diagonal matrix with $\tilde{d}_{ii} = \sum_{j=1}^N \tilde{w}_{ij}$.

As we already point out, since the assumption of these traditional manifold learning algorithms is that the embedded manifold is isometric to Euclidean space, the similarity between any two neighborhood points obviously would be over-estimated if the manifold is highly curved. Therefore, our method proposes to add the local curvature information on manifold learning.

3.2. Curvature estimation

In a continuous manifold, if there exists a cone point, the Riemannian curvature at this point tends to infinity. Since manifold would blow up in this point, we call this point as a singular point. As discussed above, the Riemannian curvature of the sub-manifold that we mentioned in this paper is captured by the Hessian matrix of the embedding map. Thus based on this method, the singularities would not appear in our proposed method. Hessian matrix is a square matrix of second-order derivatives with respect to all of the variables of a scalar-valued function (the embedding map) and it represents the concavity, convexity and the local curvature of this function. Suppose $f^\alpha(x^1, x^2, \dots, x^d)$ is a multivariable function with d parameters. Then the Hessian matrix H of f^α is given as: $H_{ij}(f^\alpha) = \frac{\partial^2 f^\alpha}{\partial x^i \partial x^j}$.

In each local patch U_i of x_i , we choose a set of local natural orthogonal coordinates $\{\frac{\partial}{\partial x^1}, \frac{\partial}{\partial x^2}, \dots, \frac{\partial}{\partial x^d}\}$. In practice, we use PCA [13] to estimate the local orthogonal coordinate system of U_i . The corresponding normal coordinates of normal space are computed by Gram-Schmidt orthogonal method. The corresponding local coordinates of $\{x_{i_1}, \dots, x_{i_K}\} \in U_i$ under this new local normal coordinate system are represented as $\{u_{i_1}, u_{i_2}, \dots, u_{i_K}\}$. x_i is projected into the original point. The second fundamental form coefficients are estimated as $h_{ij}^\alpha = \frac{\partial^2 f^\alpha}{\partial x^i \partial x^j}$, $\alpha = 1, \dots, D-d$.

Consider the Taylor expansion of f^α , $\alpha = 1, \dots, D-d$, at x_i under this new local coordinate system:

$$f^\alpha(u_{ij}) = f^\alpha(0) + u_{ij} \nabla f^\alpha + u_{ij} H^\alpha u_{ij}^T + o(\|u_{ij}\|^2). \quad (7)$$

For each component $h_{ij}^\alpha = \frac{\partial^2 f^\alpha}{\partial x^i \partial x^j}$ of Hessian matrix H^α , it can be considered as the quadratic coefficient of this quadratic polynomial function f^α . For local tangent space T_{x_i} , the orthogonal coordinate system is spanned by $\{\frac{\partial}{\partial x^1}, \frac{\partial}{\partial x^2}, \dots, \frac{\partial}{\partial x^d}\}$. For local quadratic polynomial vector space Q_{x_i} , the local coordinate system is spanned by $\{\frac{\partial}{\partial x^1}, \dots, \frac{\partial}{\partial x^d}, \frac{\partial^2}{\partial x^1 \partial x^1}, \dots, \frac{\partial^2}{\partial x^1 \partial x^2}, \dots, \frac{\partial^2}{\partial x^{d-1} \partial x^d}\}$. So the Hessian matrix H^α is estimated by projecting the input data points into the polynomial vector space Q_{x_i} . We use the least square estimation method to compute the projecting coefficients. The solution is obtained as: $B_i = \Psi^\dagger f$, where $\Psi_{ij} = [1, u_{i_1}^1, \dots, u_{i_1}^d, (u_{i_1}^1)^2, \dots, (u_{i_1}^d)^2, (u_{i_1}^1 \times u_{i_1}^2), \dots, (u_{i_1}^{d-1} \times u_{i_1}^d)]$, $\Psi = [\Psi_{i_1}, \dots, \Psi_{i_K}]$, $\Psi^\dagger$ is the pseudo-inverse matrix of Ψ and $f = [f^1, f^2, \dots, f^{D-d}]$, $f^\alpha = [f^\alpha(u_{i_1}), \dots, f^\alpha(u_{i_K})]^T$, $\alpha = 1, 2, \dots, D-d$. The learnt local projection coordinates of each point $x_{ij} \in U_i$ are given as $b_{ij} = [1, \tau_{ij}, H^{ij}]$ where τ_{ij} is the tangent components vector and H^{ij} is the vector-form representation of Hessian matrix. x_i is projected into the original point expressed as $b_{i_0} = 0$.

3.3. Curvature-aware manifold learning

In LTSA [8], the authors analyzed the reconstruction error in theory and obtained that the error is highly influenced by the curvature of sub-manifold $\mathcal{M}$. If the sub-manifold is highly curved in the higher dimensional ambient space, the reconstruction error would be very high. As we know, the accurate determination of local tangent space is dependent on several factors: curvature information embedded in the Hessian matrices, local sampling density, and noise level of data points. Thus, for LTSA, it is necessary to analyze the curvature information of sub-manifold during dimension reduction but no further study has been done in our knowledge.

We propose to design a new algorithm to improve traditional manifold learning algorithms by adding curvature information. In this paper, we focus on improving two algorithms LLE and LEP in detail and give the detailed theoretical analysis of these two

improved algorithms CA-LLE and CA-LEP. For local structure preserved method, we just divide the sub-manifold $\mathcal{M}$ into a set of local patches $\{U_i\}$. Here U_i is the local patch with respect to x_i and we choose U_i as an example to analyze our algorithm. Given a local patch U_i , $\{\frac{\partial}{\partial x^1}, \dots, \frac{\partial}{\partial x^d}, \frac{\partial^2}{\partial x^1 \partial x^1}, \dots, \frac{\partial^2}{\partial x^1 \partial x^2}, \dots, \frac{\partial^2}{\partial x^{d-1} \partial x^d}\}$ span the local polynomial vector space Q_{x_i} . By projecting original input data points $x_{ij} \in U_i$ to this local polynomial vector space respectively, we obtain the corresponding projection coefficients shown as $b_{ij} = [1, \tau_{ij}, H^{ij}]$, $j = 1, 2, \dots, K$. The local curvature information of U_i at x_{ij} is hidden in the quadratic component vector H^{ij} .

In the following, we give the detailed description of our CAML algorithm.

CAML Algorithm Procedures:

1. Input a set of data points $\{x_1, x_2, \dots, x_N\}$, $x_i \in \mathbb{R}^D$. This step is the same as the first step of traditional manifold learning algorithm. As we have known, for each i , U_i is the K -nearest neighbor set of x_i based on Euclidean metric and the sub-manifold $\mathcal{M}$ is supposed to be divided into the set of local patches $\{U_1, U_2, \dots, U_N\}$.
2. Unlike traditional manifold learning algorithms to project local patches into local tangent spaces, we project the local patch U_i into a second-order polynomial vector space and obtain the new local coordinate representations $\{b_{i_1}, b_{i_2}, \dots, b_{i_K}\}$ by Eq. (7), where $b_{ij} = [1, \tau_{ij}, H^{ij}]$. The curvature information at each point x_{ij} is hidden in H^{ij} .
3. To add the curvature information into the weight matrix W_i for each U_i , we use the local representations $\{b_{i_1}, b_{i_2}, \dots, b_{i_K}\}$. To give each element $w_{i_0 i_j}$ of the weight matrix W_i , for LEP, see Eq. (8), and for LLE, see Eq. (9) below.
4. After constructing the curvature-aware weight matrix W , we use this weight matrix to reconstruct the representations $Y = \{y_1, y_2, \dots, y_N\}$ in lower dimensional Euclidean space $\mathbb{R}^d$. Y is learnt by minimizing $\Phi(Y)$ given by Eq. (6) under some normalization constraints. This step is the same as the third step of traditional manifold learning.

To define each element of weight matrix W to LEP and LLE, we

Algorithm 1 Curvature-aware Manifold Learning.

Input: Training data points $\{x_1, x_2, \dots, x_N\}$, $x_i \in \mathbb{R}^D$, neighbor-size parameter K .

Output: $\{y_1, y_2, \dots, y_N\}$, $y_i \in \mathbb{R}^d$.

1. **for** $i = 1$ **to** N **do**
2. Find K -nearest neighbors of x_i ;
3. **end for**
4. Determine the intrinsic dimension d of $\mathcal{M}$.
5. **for** $i = 1$ **to** N **do**
6. Compute vector space B_i by Eq. (7).
7. Construct the local weight matrix W_i .
8. Eq. (8) for CA-LEP; Eq. (9) for CA-LLE.
9. **end for**
10. Minimize the reconstruction error function Eq. (6).

use the following two equations.

For Curvature-aware LEP:

$$w_{st} = \begin{cases} \exp\left(-\frac{\|b_s - b_t\|^2}{2\sigma^2}\right), & x_s, x_t \in U_i \\ 0, & x_s \notin U_i \text{ or } x_t \notin U_i \end{cases} \quad (8)$$

The global weight matrix W on all the data points is a symmetric matrix $W = [w_{st}]$, $s, t = 1, 2, \dots, N$, and W_i is a sub-block matrix of W , $W|_{U_i} = W_i$.

For Curvature-aware LLE:

$\{w_{i_0 i_j}\}$ in local patch U_i is obtained by minimizing the following equation:

$$\operatorname{argmin} \|b_{i_0} - \sum_{j=1}^K w_{i_0 i_j} b_{i_j}\|^2. \quad (9)$$

When we use b_{i_j} to denote the local coordinate representation of x_{i_j} , the curvature information of each local patch is added into the local weight matrix W_i . The detailed theoretical analysis is shown in the following section.

4. Algorithm analysis

We just consider one local patch U_i as an example to give the analysis of our curvature-aware manifold learning algorithm. Different from traditional locally preserved MAL algorithms, our method projects the original data points into a local polynomial vector space. The corresponding local projection of $x_j \in U_i$ is shown as $b_j = [1, \tau_j, H^j]$ and x_i is projected into $b_i = 0$.

4.1. Curvature-aware LEP

In the polynomial vector space, the weight value w_{ij} between two neighbor points $x_i, x_j \in U_i$ is given by:

$$w_{ij} = \exp^{-\frac{\|b_i - b_j\|^2}{2\sigma^2}} = \exp^{-\frac{\|\tau_i - \tau_j\|^2}{2\sigma^2}} \cdot \exp^{-\frac{\|H^i - H^j\|^2}{2\sigma^2}}, \quad (10)$$

where $\|H^i - H^j\|^2 = \|H^{x_j}\|_F^2$. H^{x_j} represents the Hessian matrix at point x_j . Under this new local normal coordinate frame of U_i , the coordinate of x_i is $b_i = 0$, so $H^i = 0$, obviously $\|H^j\|^2 = \|H^{x_j}\|_F^2$.

Hessian matrix is a symmetric matrix. We do the eigenvalue decomposition to H^{x_j} and obtain the following expression:

$$\|H^{x_j}\|_F^2 = \|U^T \Lambda_j U\|_F^2 = \|\Lambda_j\|_F^2, \quad (11)$$

where Λ_j is the eigenvalue matrix of H^j . In Riemannian geometry, each eigenvalue of the Hessian matrix H represents a principal curvature along the corresponding coordinate. Based on the above analysis, the weight value w_{ij} in Eq. (10) is shown as:

$$w_{ij} = \exp^{-\frac{\|\tau_i - \tau_j\|^2}{2\sigma^2}} \cdot \exp^{-\frac{\|\Lambda_j\|_F^2}{2\sigma^2}}. \quad (12)$$

It is equivalent to add a curvature penalty on the similarity weight matrix W . The higher the curvature of local patch, the smaller the weight values will be among neighborhood points.

Theorem 4.1. Assume the reconstruction error under our curvature-aware weight matrix is represented as E . And the reconstruction error under traditional LEP [6] is represented as $\tilde{E}$. Then, we have

$$\|E\|_F \leq \|\tilde{E}\|_F. \quad (13)$$

Proof. The element of weight matrix W under our CA-LEP is as defined in Eq. (12):

$$w_{ij} = \exp^{-\frac{\|\tau_i - \tau_j\|^2}{2\sigma^2}} \cdot \exp^{-\frac{\|\Lambda_j\|_F^2}{2\sigma^2}}.$$

And the weight $\tilde{W}$ under traditional LEP algorithm is defined as:

$$\tilde{w}_{ij} = \exp^{-\frac{\|\tau_i - \tau_j\|^2}{2\sigma^2}}.$$

Obviously we have $w_{ij} \leq \tilde{w}_{ij}$. The corresponding Laplace matrices are defined as $L = D - W$, $\tilde{L} = \tilde{D} - \tilde{W}$. Therefore, we have:

$$\lambda_l(L) \leq \lambda_l(\tilde{L}), l = 1, 2, \dots, N,$$

where N is the number of input data points.

For LEP, the lower dimensional representations are obtained from the d eigenvectors of the smallest d eigenvalues of Laplace

matrix. The reconstruction error E is measured by the values of the smallest d eigenvalues $\lambda_l, l = 1, 2, \dots, d$,

$$\|E\|_F = \sum_{i=1}^N \|x_i - f(y_i)\| = \sum_{l=1}^d \lambda_l.$$

We have proved that the eigenvalue of L is less than that of $\tilde{L}$. So we have

$$\|E\|_F \leq \|\tilde{E}\|_F.$$

Therefore, when considering the curvature information of sub-manifold, the reconstruction error gets much lower during dimension reduction. $\square$

4.2. Curvature-aware LLE

In each local patch U_i , we compute the local linear combination structure by minimizing the following equation:

$$\Phi_i = \|b_i - \sum_j w_{ij} b_j\|^2, \quad (14)$$

where $x_j \in U_i, b_j = [1, \tau_j, H^j], \sum_{j=1}^K w_{ij} = 1$.

The equation Φ_i in Eq. (14) can be rewritten as:

$$\Phi_i = \|\tau_i - \sum_j w_{ij} \tau_j\|^2 + \|H^i - \sum_j w_{ij} H^j\|^2. \quad (15)$$

For traditional LLE, the authors just minimized the first item of Φ_i . For our method, we add the second item to measure the linear combination of Hessian matrices (see Eq. (15)).

In the following, we give a theoretical derivation to explain the necessity of adding the second Hessian item of Φ_i .

First we give the Taylor expansion of the embedding map f in the local patch U_i :

$$f(u) = f(0) + u^T \nabla f + \frac{1}{2} (u^T H u) + o(\|u\|^2), \quad (16)$$

Under the Taylor expansion of f , we obtain the linear relationship between x_i and its rest neighbors:

$$f(0) - \sum_j w_{ij} f(u_j) \approx f(0) - \sum_j w_{ij} f(0) - \sum_j w_{ij} u_j^T \nabla f - \frac{1}{2} \sum_i w_{ij} u_j^T H u_j. \quad (17)$$

Since $\sum_j w_{ij} = 1, u_i = 0, \sum_j w_{ij} u_j = 0$,

$$f(0) - \sum_j w_{ij} f(u_j) \approx -\frac{1}{2} \sum_j w_{ij} u_j^T H u_j, \quad (18)$$

where $\sum_j w_{ij} u_j^T H u_j = \sum_j w_{ij} H^j$.

We have stated that the coordinate of x_i under this local normal coordinate frame is zero, so the corresponding Hessian matrix $H^i = 0$. Therefore Eq. (18) can be given as:

$$f(0) - \sum_j w_{ij} f(u_j) \approx \frac{1}{2} H^i - \frac{1}{2} \sum_j w_{ij} H^j. \quad (19)$$

In summary, for our method, it is necessary to add a Hessian item when constructing the local linear combination structure of each local patch U_i .

4.3. Time complexity analysis

In this subsection, we give the time complexity analysis of our CAML compared with traditional manifold learning algorithms based on the number of data points N , the input dimension $\mathcal{D}$ and the intrinsic dimension d . Comparing with traditional manifold learning algorithms, the added time cost of our method mainly focuses on the computation of Riemannian curvature information of datasets. The main process of this step is to estimate the local

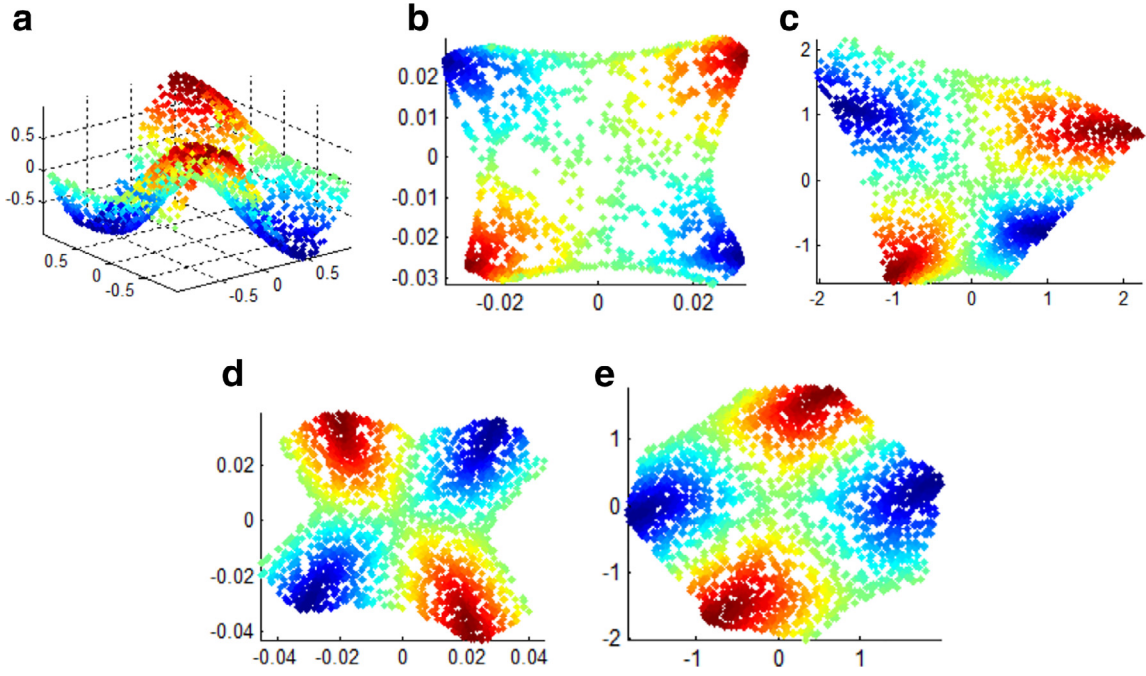


Fig. 1. Twin Peaks dataset embedding by four algorithms. (a) the original dataset, (b) LEP algorithm, (c) LLE algorithm, (d) CA-LEP, (e) CA-LLE.

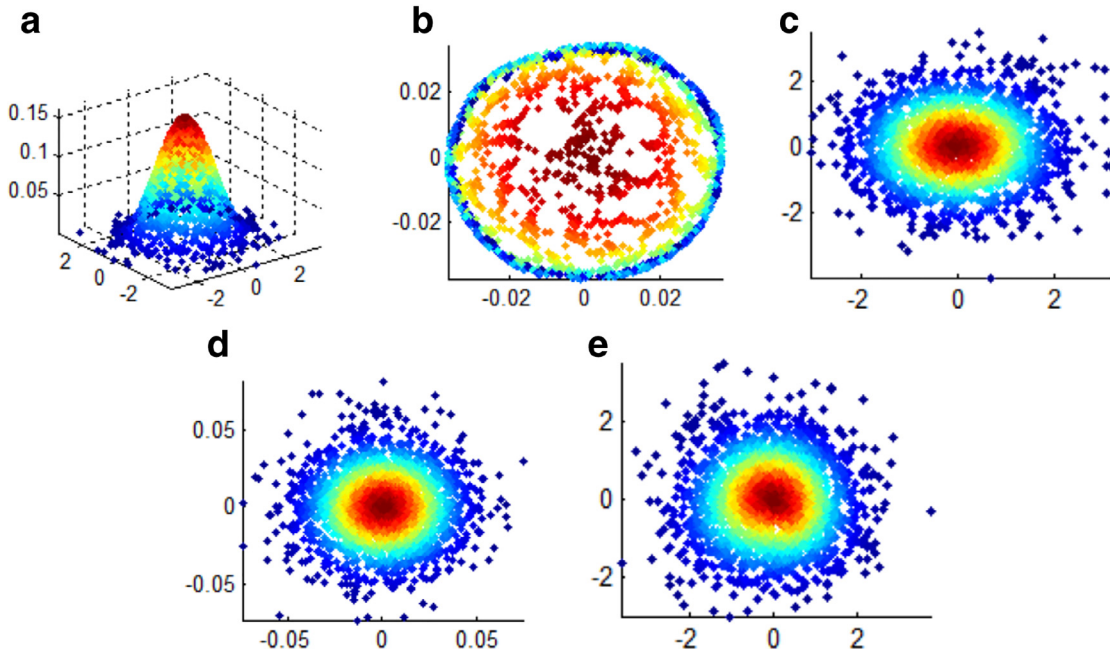


Fig. 2. Gaussian dataset embedding by four algorithms. (a) the original dataset, (b) LEP algorithm, (c) LLE algorithm, (d) CA-LEP, (e) CA-LLE.

analytical structure by fitting a second-order polynomial function shown in Eq. (7). The Riemannian curvature of each local patch is obtained by computing the eigenvalues of the corresponding Hessian matrix, where the size of each Hessian matrix is $d \times d$. The time cost of eigenvalue decomposition to each Hessian matrix is $O(d^2)$, hence the total time cost of the full samples is shown as $O(Nd^2)$. In general, the intrinsic dimension d is far less than the input dimension $\mathcal{D}$. In addition, only the time cost of finding K nearest neighbors of all samples is $O(N^2(\mathcal{D} + K))$ which is especially higher than the added time cost of our method.

In short, comparing with the total time cost of traditional MAL algorithms, the time cost of CAML is slightly higher than them. If

the number of samples N is especially large, the added time cost of CAML can be ignored.

5. Experiments

In this section, we compare CAML with several traditional MAL algorithms on four synthetic databases e.g. *Swiss Roll*, *Punctured Sphere*, *Gaussian*, and *Twin Peaks* [23] as well as two real world datasets e.g. Extended YaleFace B Database and USPS Database. For synthetic databases, we respectively show the effectiveness of our algorithm on two tasks: dimension reduction and parameter sensitivity analysis. For real world datasets, we compare the classification performance of our algorithm with other related methods.

Table 2

The neighborhood preserving ratio (NPR) comparisons of our CA-LEP and CA-LLE algorithms with other eight traditional manifold learning algorithms under four datasets (Twin Peaks, Swiss Roll, Punctured Sphere, and Gaussian).

Methods	Twin Peaks	Swiss Roll	Punctured Sphere	Gaussian
MDS [10]	0.6968	0.4352	0.6774	0.9082
PCA [13]	0.5567	0.4167	0.4941	0.8960
LEP [6]	0.6841	0.2145	0.7049	0.5400
LLE [3]	0.7352	0.6156	0.5362	0.8912
IsoMap [2]	0.8259	0.7957	0.5516	0.8568
Hessian [9]	0.7168	0.5739	0.4263	0.9073
LTSA [8]	0.8148	0.5143	0.4593	0.8960
Diffusion Map [25]	0.5576	0.2290	0.4863	0.8962
CA-LEP	0.8516	0.2739	0.7936	0.9002
CA-LLE	0.8365	0.6604	0.8040	0.9475

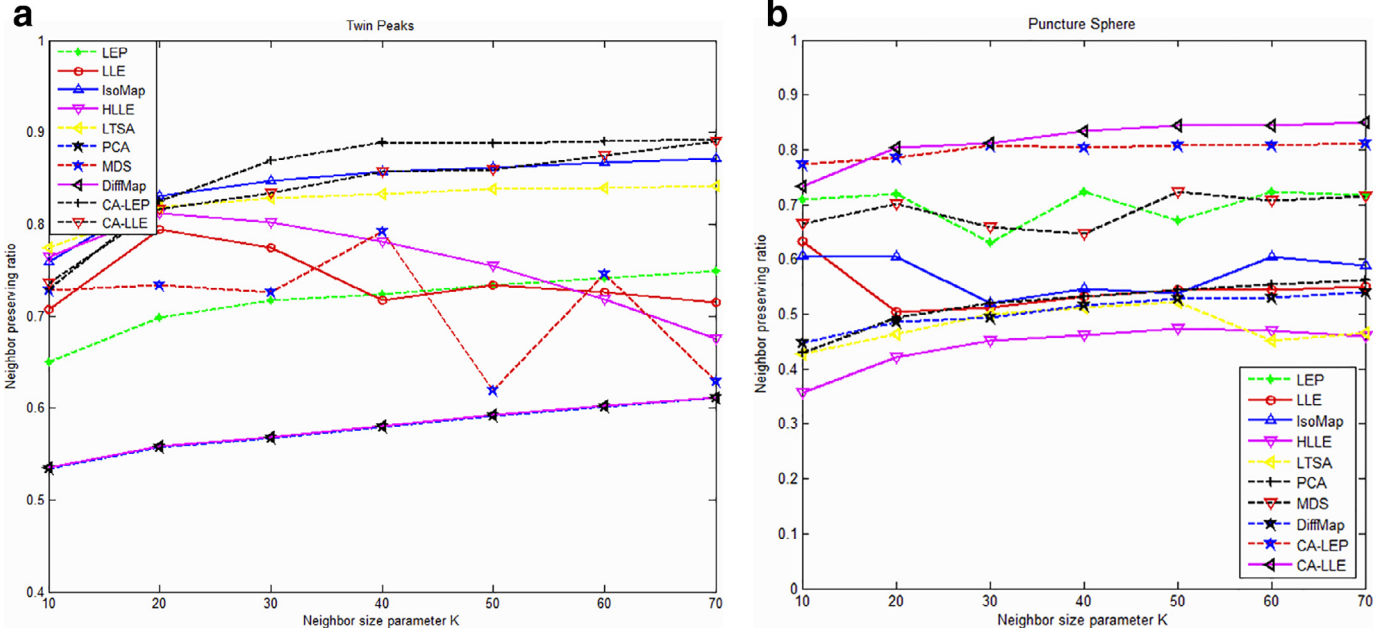


Fig. 3. Neighbor preserving ratio (NPR) vs. neighbor-size parameter K on Twin Peaks dataset and Punctured Sphere dataset.

5.1. Topology structure

Before executing dimension reduction, we first analyze the topology structures of these four synthetic databases. All the datasets are generated from Matlab code ‘mani.m’ [23]. For each dataset, it contains 2000 points distributed on the corresponding synthetic manifold. Swiss Roll is a locally flat manifold which is locally isometric to Euclidean space. For this dataset, traditional manifold learning algorithms can uncover the intrinsic structure of it accurately. For Punctured Sphere dataset, these 2000 data points lie on a two dimensional sphere which is embedded in $\mathbb{R}^3$. The curvature of this sphere is non-zero everywhere, so it is not locally or globally isometric to Euclidean space. Twin Peaks manifold is a highly curved two-dimensional manifold embedded in $\mathbb{R}^3$. Two dimensional Gaussian manifold is also not isometric to Euclidean space, where the Gauss curvature of Gaussian manifold is not zero everywhere. Therefore, traditional MAL algorithms cannot accurately uncover the intrinsic structure of these three curved synthetic manifolds except for Swiss Roll.

Based on the analysis of these four synthetic manifolds, we compare our curvature-aware manifold learning algorithm with eight traditional MAL algorithms in the next two subsection to show the necessity of using curvature information.

5.2. Dimension reduction

In this subsection, to fully evaluate the performance of our curvature-aware algorithm, we compare our method with eight traditional MAL algorithms (e.g. MDS, PCA, IsoMap, LLE, LEP, HLLE, DFM, LTSA). The objective of this comparison is to analyze the neighborhood preserving ratio (NPR) [24] of different algorithms during dimension reduction. All algorithms mentioned in this experiment map these four datasets into two dimensional space. To robustly evaluate the performance of different methods to different neighbor-size parameter values, we repeat the experiment 10 times for each method. Table 2 shows the average NPRs of each method, where the neighbor-size parameter K is taken in the set of $\{10, 20, \dots, 100\}$. The neighborhood preserving ratio (NPR) is defined as:

$$NPR = \frac{1}{KN} \sum_{i=1}^N |\mathcal{N}(x_i) \cap \mathcal{N}(y_i)|. \quad (20)$$

x_i represents the input data point and y_i is the corresponding low dimensional representation. $\mathcal{N}(x_i)$ is the set of subscripts $\{j\}$, where x_j is the K -nearest neighbor of x_i , and the same $\mathcal{N}(y_i)$ is the set of $\{l\}$, where y_l is the K -nearest neighbor of y_i . $|\cdot|$ represents the number of intersection points.

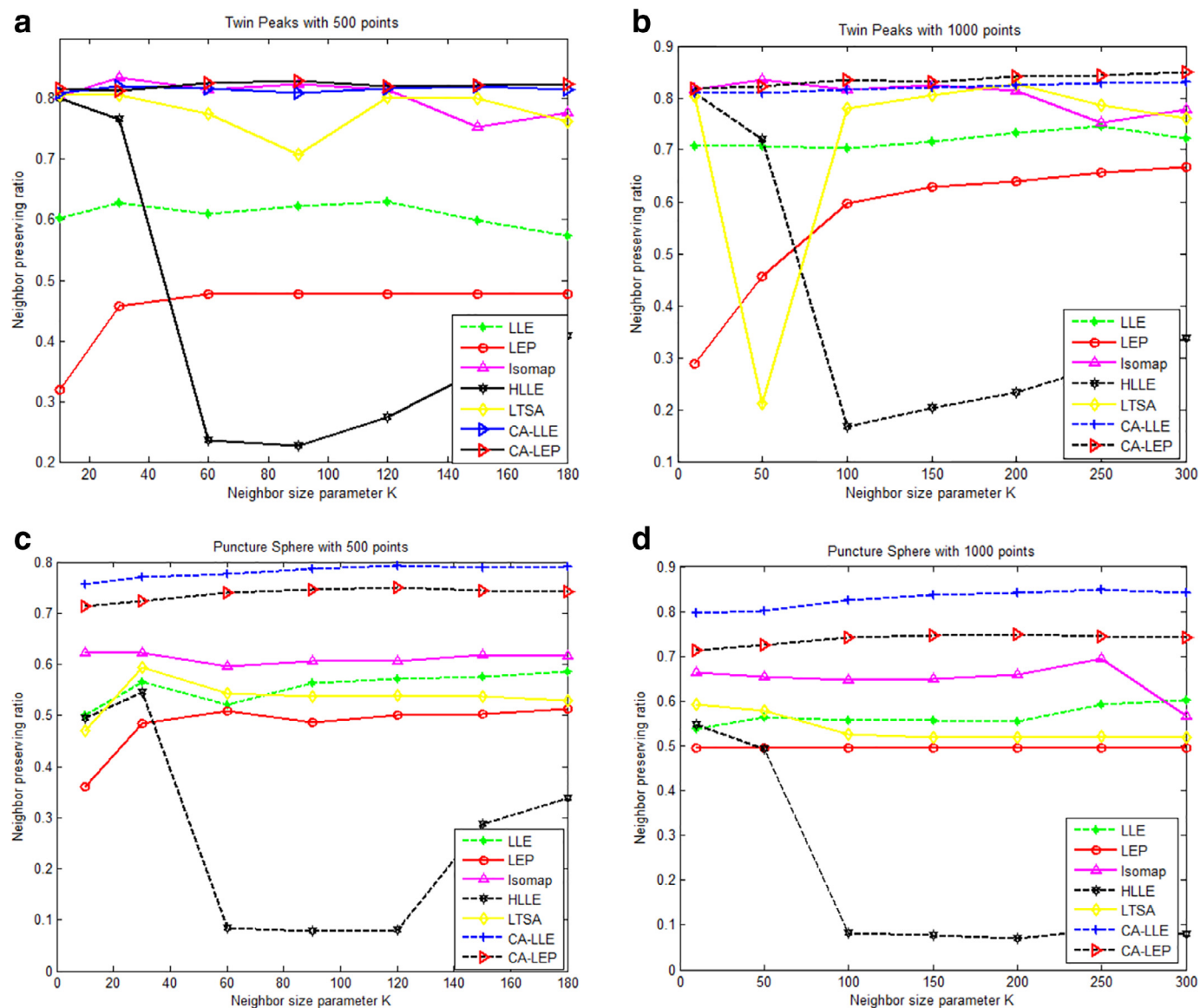


Fig. 4. Neighbor preserving ratio (NPR) vs. the density of two synthetic datasets (Twin Peaks dataset and Puncture Sphere dataset).

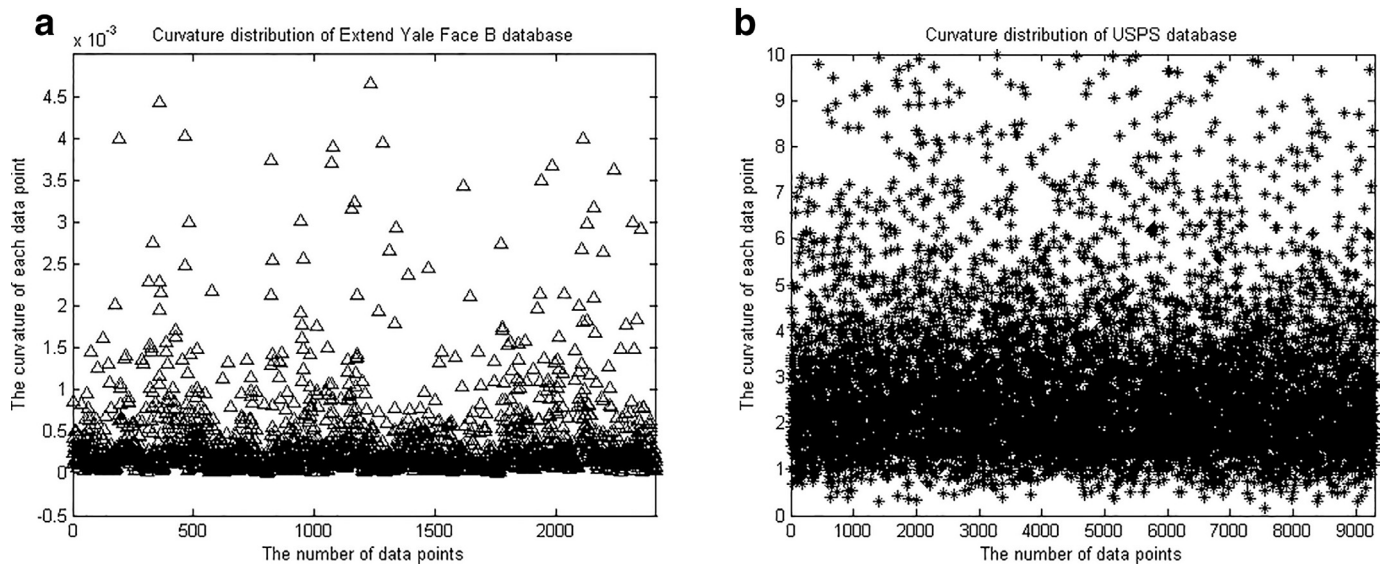


Fig. 5. The curvature distributions of Extended YaleFace B database (a) and USPS database (b).

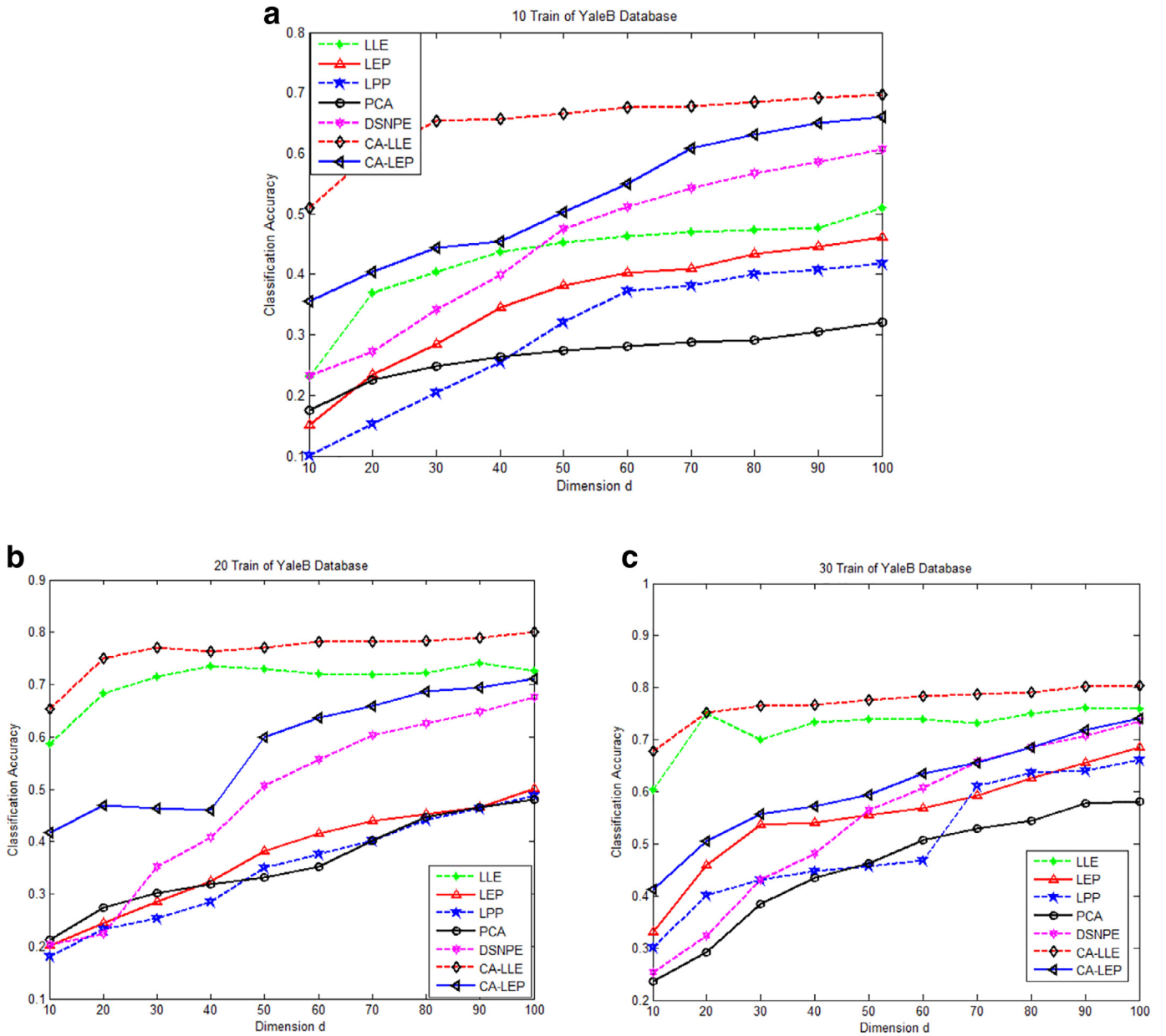


Fig. 6. Classification accuracy vs. dimensionality on Extended YaleFace B database with 10, 20, 30 respectively images per subject for training.

Table 2 shows that for all datasets except Swiss Roll, the NPRs of our CA-LEP and CA-LLE are higher than the rest traditional MAL algorithms. Swiss Roll is a flat Riemannian manifold, so our algorithm has almost no advantages on this dataset. For Gaussian dataset, it is a symmetric and convex manifold. So the NPRs of all algorithms are much higher. For Punctured Sphere and Twin Peaks, the NPRs of our algorithm are obviously more accurate than those from traditional MAL algorithms. These results clearly demonstrate that our CAML algorithm is more stable and better to discover the local structure of data points.

In order to better understand the embedding process, we give two intuitive examples shown in Figs. 1 and 2. These two figures respectively illustrate the intuitive embedding results of Twin Peaks dataset and Gaussian dataset under four different dimension reduction algorithms. From Fig. 2, we can see that since the special distribution of Gaussian dataset, the embeddings of LLE and CAML are highly similar and structure preserved well.

5.3. Sensitivity analysis

As we have shown, traditional MAL algorithms are sensitive to some parameters e.g. neighbor-size parameter K , intrinsic dimension d , and density of data points. However, for intrinsic dimension d , we have not a very suitable method to estimate it exactly. So, in this paper, we assume that the intrinsic dimension d of sub-manifold $\mathcal{M}$ is unique and approximately estimated [26]. In this experiment, we mainly analyze the sensitivity to neighbor-size parameter K and the robustness to scarce data.

5.3.1. Parameter sensitivity analysis

We compare the NPRs of different manifold learning algorithms w.r.t different neighbor-size parameter values K , $K = 10, 20, 30, 40, 50, 60, 70$, respectively. All the experiments are executed on these four datasets (Twin Peaks, Swiss Roll, Punctured Sphere, and Gaussian) with 2000 data points. By analyzing in the above subsection, Swiss Roll is a flat manifold and the Gaus-

Table 3

Classification performance of YFB DB, USPS DB, together with the comparison results for CAML (CA-LEP and CA-LLE) and traditional manifold learning algorithms PCA, LPP, LEP and LLE.

YFB DB	YFB-trn10/tst40	YFB-trn20/tst30	YFB-trn30/tst20	YFB-trn40/tst10
PCA [13]	24.26 ± 1.2	35.31 ± 2.8	43.72 ± 1.9	50.97 ± 1.7
LPP [7]	29.70 ± 2.5	35.54 ± 1.5	50.84 ± 1.8	58.29 ± 1.4
LEP [6]	34.43 ± 2.1	37.87 ± 1.3	54.28 ± 1.8	62.62 ± 1.5
LLE [3]	40.01 ± 1.9	69.72 ± 1.8	70.56 ± 2.3	71.42 ± 1.7
CA-LEP	50.03 ± 1.4	58.02 ± 1.2	63.38 ± 1.6	69.87 ± 1.3
CA-LLE	61.21 ± 1.5	75.31 ± 0.9	78.64 ± 1.3	79.39 ± 1.2
USPS DB	USPS-trn300/tst400	USPS-trn400/tst300	USPS-trn500/tst200	USPS-trn600/tst100
PCA [13]	86.68 ± 1.3	84.71 ± 1.2	86.40 ± 1.5	87.20 ± 1.1
LPP [7]	86.45 ± 1.4	88.62 ± 2.1	89.84 ± 1.3	90.14 ± 1.5
LEP [6]	91.25 ± 1.2	91.89 ± 0.8	92.03 ± 0.7	92.81 ± 1.3
LLE [3]	89.72 ± 1.3	90.97 ± 1.2	91.30 ± 0.8	92.48 ± 0.6
CA-LEP	93.31 ± 1.1	93.52 ± 1.6	94.14 ± 1.4	94.52 ± 1.2
CA-LLE	91.46 ± 0.9	92.08 ± 1.3	92.31 ± 1.7	93.04 ± 1.5

sian curvature is equal to 0 everywhere. Therefore, we just analyze the effectiveness of our proposed curvature-aware method on the other three data sets. In order to highlight the improvement of our algorithm, we still use eight traditional MAL algorithms (mentioned above) to compare with CAML. Since the distribution of Gaussian dataset is a symmetric convex structure and the Gaussian curvature on each point is approximately equal to zero, all the traditional MAL algorithms are very insensitive to the neighbor-size parameter K . In addition, for all the MAL algorithms except LEP, the neighborhood preserving ratios tend to 1 under different neighbor-size parameter values. Several intuitive illustrations have been shown in Table 2 and Fig. 2. Thus, we just give the experimental analysis on Twin Peaks dataset and Puncture Sphere dataset to analyze the sensitivity of all the methods to the neighbor-size parameter K .

The final comparison results are shown in Fig. 3. Easy to see that, our method outperforms the existing MAL algorithms when $K \geq 20$. In addition, from Fig. 3(a) and (b) we can see that the NPR curves of these traditional MAL algorithms are very sensitive to neighbor-size parameter K , while the NPRs of CA-LEP and CA-LLE are steady growth as the increase of neighbor-size parameter K . Focus on the comparison between the traditional LEP, LLE and our CALEP, CALLE, one can see that our method outperforms them in any case.

5.3.2. Robust analysis

In this section, we propose to analyze the robustness of our method to scarce data. We respectively generate 500 points and 1000 points from Twin Peaks database and the same from Puncture Sphere database. In the dimension reduction process, we respectively choose different neighbor-size parameter K . Then, we compute the neighborhood preserving ratio under fixed neighbor-size $K = 30$. Since PCA, MDS and Diffu-Map do not need to determine the neighbor-size parameter K , we just use five traditional MAL algorithms to compare with our proposed method. The final result is shown in Fig. 4. As can be seen, our methods outperform all the other traditional MAL algorithms in all cases. Among these methods, HLLC is the most sensitive to the density of data points. Moreover, under 500 points, the NPRs of these traditional MAL algorithms are significantly lower than those under 1000 points. However, our method is more robust to scarce data than other methods.

Above all, for synthetic database, our CAML algorithm is qualified to overcome some limitations of traditional MAL algorithms.

5.4. Real world experiments

In this experiment, we consider the application of our algorithm on two real-world datasets: Extended Yale Face B database

[27] and USPS database [28]. The main purpose of this experiment is to test the classification accuracy of different methods.

5.4.1. About the databases

The Extended YaleFace B database, or YFB DB for short, contains 2414 single light source images of 38 individuals each seen under about 64 near frontal images. For every subject in a particular pose, an image with ambient illumination is also captured. The face region in each image is resized into 32×32 , so the original dimension of this database is 1024.

The USPS database consists of 9298 images. It refers to numeric data obtained from the scanning of handwritten digits from envelopes by the U.S. Postal Service. The original scanned digits are binary and of different sizes and orientations. The images here have been size normalized, resulting in 16×16 gray-scale images, so the original dimension of this database is 256.

5.4.2. Evaluation

In this experiment, we first analyze the curvature distributions of YFB database and USPS database shown in Fig. 5. One can see that the embedded manifold of USPS database is highly curved in higher dimensional Euclidean space. The curvature value in each data point of USPS database is almost higher than 0.5. One reason is that the handwritten digits from different classes are vary greatly. Meanwhile, for Extended YaleFace B database, the curvature distribution of each point is in the range of 0 to 5×10^{-3} . It means that the local geometric structure of YFB database is close to a flat space.

In the second step of this experiment, we compare our algorithm with several traditional manifold learning algorithms under these two databases. The whole experimental process is shown as follows: Firstly, we use manifold learning algorithms to reduce the dimension of databases. Secondly, in the low dimensional space, we use *Nearest Neighbor Classifier* to test the classification accuracies of these two databases.

For YFB database, we choose 50 images from each class, then obtain totally 1900 images for our experiment. We totally run the experiment four times by each algorithm. In each experiment, we randomly choose $p(p = 10, 20, 30, 40)$ images per subject as the training dataset, the rest $50 - p$ images per subject as the testing dataset, respectively. The average classification accuracy results of different algorithms are shown in Table 3 on upper part, where the neighbor-size parameter K is tuned from 10 to 100 and the dimension d is fixed to 10. The main purpose of this experiment is to find how much improvement has been contributed by our curvature-aware manifold learning algorithm comparing with traditional MAL algorithms. Notice that from Table 3, the classification results of MAL algorithms mostly outperform the linear dimension reduction algorithm PCA. In addition, the classification results of

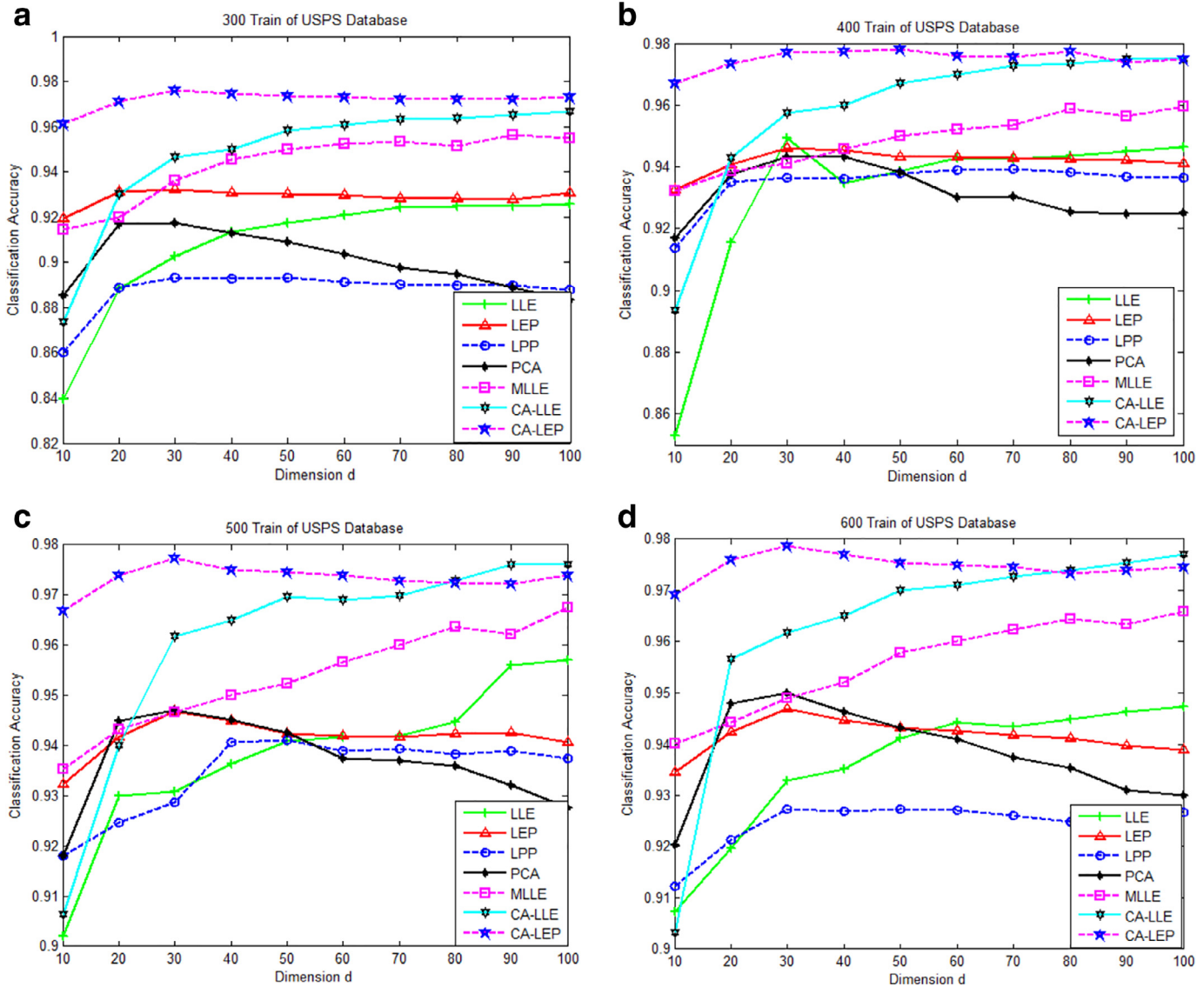


Fig. 7. Classification accuracy vs. dimensionality on USPS database with 300, 400, 500, 600 respectively images per subject for training.

LPP and LEP are especially higher than LLE. One main reason is that LLE assumes the local patches of data points being linear space and uncovers its linear combination relationship. Among all these classification results, we especially propose to analyze the comparisons between traditional MAL algorithms and our proposed algorithm. Easy to see that after adding curvature information to LLE, the classification results of CA-LLE slightly outperform LLE. One main reason is that the curvature distribution of YFB database is close to zero. Furthermore, comparing with traditional LEP, our CA-LEP algorithm obviously performs better. In all cases, the performance of our curvature-aware manifold learning algorithm is better than all the other MAL algorithms.

For USPS Database, we respectively choose 700 images per subject in this experiment. We also do the experiments four times by each algorithm respectively. Using the same method with YFB DB, we randomly choose p , ($p = 300, 400, 500, 600$) respectively image sets (per subject) for training, the rest for testing. The average classification results of different algorithms are shown in Table 3 on lower part. From these results, we can see that the classification results of traditional MAL algorithms outperform PCA in any case. For our curvature-aware manifold learning, by adding the curvature information to dimension reduction, the classification accu-

cies of our algorithm are higher than the other MAL algorithms. Among these results, it is worthy to point out that comparing with the classification accuracies between LEP, LLE and CA-LEP. CA-LLE, the results of our method significantly outperform them.

5.4.3. Qualitative analysis

In order to show a clear comparison of the effectiveness of our proposed method with the existing algorithms, we further give several qualitative results. Previously, we have already mentioned that the intrinsic dimension estimation is affected by the Riemannian curvature. Until now, the advantage of our proposed approach has not been demonstrated for dimension estimation. Here, we study some qualitative results under different low-dimension d . In the dimension reduction step, we respectively take the low-dimension d in the set of $[10, 20, \dots, 100]$ and set the neighbor-size parameter K to be 30. For YFB database, we respectively choose $p = 10, 20, 30$ images per subject for training, the rest for testing. The corresponding classification accuracy results are shown in Fig. 6. For USPS database, we choose $p = 300, 400, 500, 600$ images per subject for training, and the corresponding results are displayed in Fig. 7.

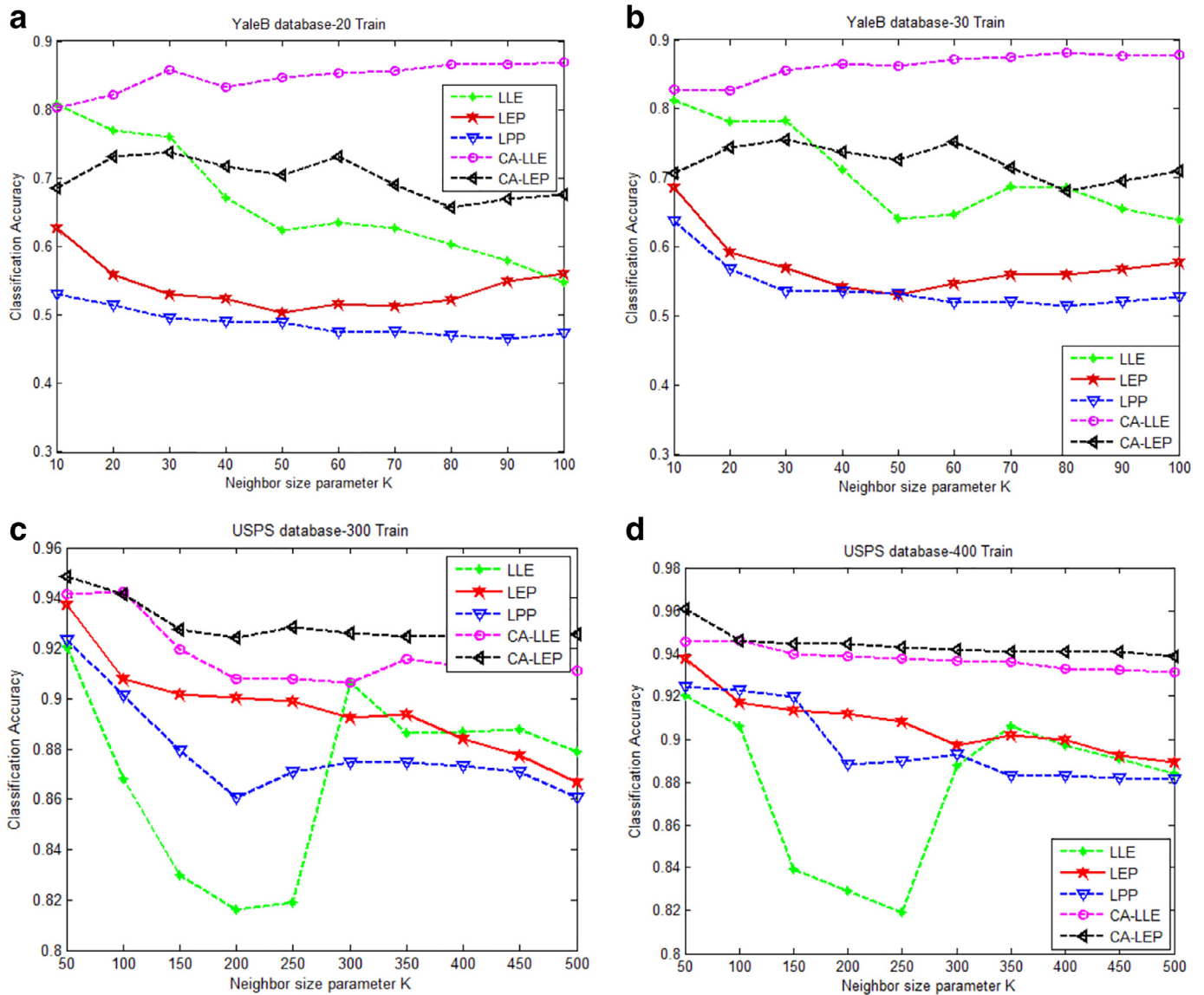


Fig. 8. Classification accuracy vs. neighbor-size parameters on YFB database and USPS database.

From YFB database experimental results, we can see that our CA-LLE achieves the highest accuracy rate in all cases. Again, among these results, focus on the comparison between LLE, LEP and our CA-LLE, CA-LEP, our method performs the best in all cases. In addition, comparing the manifold learning based algorithms with sparse representation method DSNPE [1], the results show that when the dimension $d < 100$ the accuracy rate of DSNPE is lower than our method. From [1], we can see that DSNPE would achieve high performance only when the dimension d is greater than 200. Therefore, our CAML algorithm achieves competitive results under lower dimension $d < 100$. For USPS database, compared with LLE, the classification accuracy curves of MLLE are relatively stable. CA-LLE and CA-LEP outperform all other methods in any case. One reason is that the curvature value of USPS database is almost higher than 0.5 as shown in Fig. 5.

Moreover, we do some qualitative experiments under different neighbor-size parameter K . In this experiment, the dimension d is set to be 50. For YFB database, we respectively choose 20 and 30 images per subject for training. For USPS database, we randomly use 300 and 400 images per subject as training set, the rest as test set. The final experimental result is shown in Fig. 8. As can be seen,

the accuracy curves of these traditional MAL algorithms change rapidly, while the corresponding accuracy curves of our method are relatively stable. In addition, our method outperforms all other algorithms in all cases.

In summary, when adding the curvature information of data points into manifold learning, the classification results of our algorithm outperform the traditional manifold learning algorithms in almost any case.

5.4.4. Other comparison

Besides comparing our proposed method with the existing dimensionality reduction algorithms, we also do some comparative experiments to our most related work r-Lap [18] on these two real world databases. r-Lap proposes to add curvature information to re-weight the Laplace matrix which is a regularization item of the semi-supervised learning. In the experiment of [18], r-Lap focuses on binary classification problems. Thus we transform these multi-class databases into several binary classification problems.

YFB database and USPS database respectively contain 38 classes and 10 classes. For YFB database, we randomly construct 20 binary classification problems by choosing 20 pairs of classes, and use 50

Table 4

Average binary classification performance of YFB DB, USPS DB, together with the comparison results for CAML (CA-LEP and CA-LLE) and r-Lap.

YTB DB	Classification accuracy
LLE [3]	80.25%
LEP [6]	74.35%
r-Lap [18]	78.12%
CA-LLE	86.14%
CA-LEP	84.51%
USPS DB	Classification accuracy
LLE [3]	93.86%
LEP [6]	94.14%
r-Lap [18]	95.21%
CA-LLE	96.36%
CA-LEP	97.03%

data points per subject as training set the rest as test set. For USPS database, we randomly construct 5 binary classification problems and randomly choose 200 images per subject for training. For each method, the experiment is repeated 10 times with different sets of labeled examples and the results are averaged. The corresponding total average classification results of different binary classification problems for different methods are shown in Table 4. From these experimental results, we can see that our proposed method outperforms r-Lap in all cases.

6. Conclusions and future works

To precisely describe the continuous change of point cloud, one critical step of manifold learning is to assume the dataset distributed on a lower dimensional embedded manifold. Then researchers use the mathematical theoretical knowledge of manifold to deal with these datasets, such as dimensionality reduction, classification, clustering, recognition and so on. Whether the manifold structure is uncovered exactly or not directly impacts the learning results. Traditional MAL algorithms just consider the distance metric. However, general Riemannian manifold may be not isometric to Euclidean space. So our method aims to excavate the higher order geometric quantity, Riemannian curvature, of Riemannian submanifold and uses curvature information as well as distance metric to uncover the intrinsic geometric structure of local patches. The extensive experiments have shown that our method is more stable comparing with other traditional manifold learning algorithms. The main contribution of our work presented in this paper is that it is the first time to try to add curvature information on high dimensional data points for dimensionality reduction algorithm and give the theoretical analysis.

We have pointed out that the intrinsic dimension estimation is affected by the curvature of the underlying manifold. However, we have not yet found a suitable method to solve this problem. In the future, we will consider this question further. And our work will try to use Ricci flow to dynamically uncover the intrinsic curvature structure of sub-manifold. Furthermore, we will look into the Ricci flow theory, find more useful theoretical techniques, and apply them in the study of manifold learning.

Acknowledgments

This work is supported by the National Key Research and Development Program of China under grant 2016YFB1000902; NSFC project Nos. 61232015, 61472412, 61621003; the Beijing Science and Technology Project: Machine Learning based Stomatology; Tsinghua-Tencent-AMSS-Joint Project: WWW Knowledge Structure and its Application.

References

- [1] J. Gui, Z. Sun, W. Jia, R. Hu, Y. Lei, S. Ji, Discriminant sparse neighborhood preserving embedding for face recognition, *Pattern Recognit.* 45 (2012) 2884–2893.
- [2] J.B. Tenenbaum, V. de Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, *Science* 290 (2000) 2319–2323.
- [3] S.T. Roweis, L.K. Saul, Nonlinear dimensionality by locally linear embedding, *Science* 290 (2000) 2323–2326.
- [4] V.D. Silva, J.B. Tenenbaum, Global versus local methods in nonlinear dimensionality reduction, *NIPS* 16 (2003) 705–712.
- [5] G. Rosman, M.M. Bronstein, A.M. Bronstein, R. Kimmel, Nonlinear dimensionality reduction by topologically constrained isometric embedding, *Int. J. Comput. Vis.* 89 (2010) 56–68.
- [6] M. Belkin, P. Niyogi, Laplacian eigenmaps and spectral techniques for embedding and clustering, *NIPS* 14 (2001) 585–591.
- [7] X. He, P. Niyogi, Locality preserving projections, *NIPS* 16 (2003) 153–160.
- [8] Z. Zhang, H. Zha, Principal manifolds and nonlinear dimension reduction via local tangent space alignment, *SIAM J. Sci. Comput.* 26 (2004) 313–338.
- [9] D.L. Donoho, C.E. Grimes, Hessian eigenmaps: locally linear embedding techniques for high-dimensional data, *Proc. Natl. Acad. Sci. USA* 100 (2003) 5591–5596.
- [10] M.A.A. Cox, T.F. Cox, Multidimensional scaling, in: *Handbook of Data Visualization*, 2008, pp. 315–347.
- [11] X. Xing, K. Wang, Z. Lv, Y. Zhou, S. Du, Fusion of local manifold learning methods, *IEEE Signal Process. Lett.* 22 (2015) 395–399.
- [12] Z. Zhang, J. Wang, Mlle: modified locally linear embedding using multiple weights, *NIPS* 19 (2006) 1593–1600.
- [13] H. Abdi, L.J. Williams, Principal component analysis, *Inc. WIREs Comp Stat* 2 (2010) 433–459.
- [14] B. Lin, X. He, C. Zhang, M. Ji, Parallel vector field embedding, *J. Mach. Learn. Res.* 14 (2013) 2945–2977.
- [15] P. Dollar, V. Rabaud, S. Belongie, Non-isometric manifold learning: analysis and an algorithm, *ICML* (2007) 241–248.
- [16] J.M. Lee, *Introduction to Smooth Manifolds*, Springer-Verlag, New York, 2003.
- [17] J.M. Lee, *Riemannian Manifolds: An Introduction to Curvature*, Springer, New York, 1997.
- [18] K.I. Kim, J. Tompkin, C. Theobalt, Curvature-aware regularization on Riemannian submanifolds, *Proceedings of the ICCV* (2013) 881–888.
- [19] W. Xu, E.R. Hancock, R.C. Wilson, Regularising the Ricci flow embedding, *SSPR/SPR* 6218 (2010) 579–588.
- [20] W. Xu, E.R. Hancock, R.C. Wilson, Ricci flow embedding for rectifying non-euclidean dissimilarity data, *Pattern Recognit.* 47 (2014) 3709–3725.
- [21] P. Petersen, *Riemannian Geometry*, Springer, New York, 1998.
- [22] Y. Goldberg, A. Zakai, D. Kushnir, Y. Ritov, Manifold learning: the price of normalization, *J. Mach. Learn. Res.* 9 (2008) 1909–1939.
- [23] T. Wittman, Manifold learning Matlab demo, <http://www.math.umn.edu/wittman/mani/>, 2005 (accessed 17.05.30).
- [24] G. Sanguinetti, Dimensionality reduction of clustered data sets, *IEEE TPAMI* 30 (2008) 535–540.
- [25] R. Coifman, S. Lafon, Diffusion maps, *Appl. Comput. Harmon. Anal.* 21 (2006) 5–30.
- [26] P. Mordohai, G. Medioni, Dimensionality estimation, manifold learning and function approximation using tensor voting, *J. Mach. Learn. Res.* 11 (2010) 411–450.
- [27] <http://vision.ucsd.edu/~iskwak/ExtYaleDatabase/ExtYaleB.html>.
- [28] <http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass.html#usps>.

Yangyang Li received the B.E. degrees in Math and computer science from the Hebei university, Hebei, China, in 2012. She is currently pursuing the Ph.D. degree in Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China. Her research interests include pattern recognition, image processing, and machine learning.