SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

September 2019, Vol. 62 192101:1–192101:14
https://doi.org/10.1007/s11432-018-9702-7

Applying Ricci flow to high dimensional manifold learning

Yangyang LI[1,2*] & Ruqian LU[1]

[1]*Key Lab of Management, Decision and Information Systems, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing* 100190*, China;*
[2]*University of Chinese Academy of Sciences, Beijing* 100049*, China*

Received 21 March 2018/Revised 28 May 2018/Accepted 11 September 2018/Published online 14 May 2019

Abstract In machine learning, a high dimensional data set such as the digital image of a human face is often viewed as a point set distributed on a differentiable manifold. In many cases, the intrinsic dimension of this manifold is low but the representation dimension of the data points is high. To ease data processing requirements, manifold learning (ML) techniques can be used to reduce a high dimensional manifold (HDM) to a low dimensional one while keeping the essential geometric properties, such as relative distances between points, unchanged. Traditional ML algorithms often assume that the local neighborhood of any point on an HDM is roughly equal to the tangent space at that point. This assumption leads to the disadvantage that the neighborhoods of points on the manifold, though they have a very different curvature, will be treated equally and will be projected to a lower dimensional space. The curvature is a different way of manifold processing, where traditional dimension reduction is ineffective at preserving the neighborhood. To overcome this obstacle, we perform an "operation" on the HDM using Ricci flow before a manifold's dimension reduction. More precisely, with the Ricci flow, we transform each local neighborhood of the HDM to a constant curvature patch. The HDM, as a whole, is then transformed into a subset of a sphere with constant positive curvature. We compare the proposed algorithm with other traditional manifold learning algorithms. Experimental results have shown that the proposed method outperforms other ML algorithms with a better neighborhood preserving rate.

Keywords manifold learning, Ricci flow, Ricci curvature, dimension reduction, curvature estimation

Citation Li Y Y, Lu R Q. Applying Ricci flow to high dimensional manifold learning. Sci China Inf Sci, 2019, 62(9): 192101, https://doi.org/10.1007/s11432-018-9702-7

1 Introduction

In machine learning tasks, one common problem is the redundant dimensions of data representation. Manifolds usually arise from data generated in a continuous process. The generated manifold is often embedded in a high-dimensional Euclidean space. In most cases, the manifold is represented as a discrete data set. An intuitive example of this is the set of images generated by a continuously changing set of facial expressions. This set of data points can be accurately represented by a low dimensional set of features. To uncover these features, Roweis et al. [1] and Tenenbaum et al. [2] introduced a new research field called manifold learning (in this paper, manifold learning only refers to a nonlinear dimensionality reduction technique). This technique is a perfect combination of classical geometric analysis and computer science with a detailed analysis shown in [3]. Traditional manifold learning algorithms aim to reduce the dimension of a manifold, so that its lower dimensional representation (i.e., the features) can

* Corresponding author (email: liyangyang12@mails.ucas.ac.cn)

reflect the intrinsic geometric and topologic structure of the high dimensional data points. Generally, the existing manifold learning algorithms can be roughly divided into two classes: one preserving the global geometric structure of the manifold, such as Isomap [2] and the recently incremental method for Isomap (topology learning embedding, TLE) [4], while the other one is to preserve the local neighborhood geometric structure, such as LLE (locally linear embedding) [1], LEP (Laplacian eigenmap) [5], LPP (locality preserving projections) [6], Lie-LPP [7], LTSA (local tangent space alignment) [8], Hessian Eigenmap [9]. Isomap aims to preserve the geodesic distance between any two high dimensional data points, which can be viewed as an extension of multidimensional scaling (MDS) [10]. Local neighborhood preserving algorithms approximate manifolds with a union of locally linear patches (possibly overlapped). After the local patches are estimated with linear methods such as principal component analysis (PCA) [11], the global representation is obtained by aligning the local patches together. Manifold learning algorithms have been applied to many applications such as data dimension compression, computer vision, and image recognition, among others. Despite the success of manifold learning, several problems still remained:

• **Locally short circuit problem.** If the embedded manifold is highly curved, the Euclidean distance between any two points is obviously shorter than the intrinsic geodesic distance. In this case the geodesic distance between these two points is often underestimated.

• **Intrinsic dimension estimation problem.** Since tangent spaces are simply taken as local patches, the intrinsic dimension of the manifold cannot be determined by the latter, in particular, in the case of a strongly varying curvature.

• **Curvature sensitivity problem.** If the curvature of the original manifold is especially high at some point, very small size patches are needed to represent the neighborhoods around this point. But there may not be as many data points as needed to produce enough small patches, especially when the data points are sparse.

Among the above mentioned three problems, the third one is the most critical. It is the basis of the other two. The main focus of this paper is to solve this problem.

1.1 Motivation

In manifold learning, an irregularly curved manifold is usually directly mapped to a lower dimensional Euclidean space. This is often impractical since attention is not paid to the varying geometric structure of the manifold at different points. To avoid this problem, some researchers regard the dataset as being sampled from multiple manifolds and not just from a simple manifold, such as Refs. [12, 13]. However, these approaches still cannot solve the local nonlinear structure of a manifold. Ricci flow, also known as intrinsic curvature flow, is a very useful tool for evolving an irregular manifold and making it converge to a regular one, i.e., constant curvature manifold. It has been used to prove the Poincare conjecture [14]. In this study, we use the Ricci flow to regularize the metric and curvature of the generated manifold before reducing its dimension. The proposed algorithm, namely Ricci flow manifold learning (RF-ML) algorithm, transforms the irregular manifold to a regular constant curvature manifold. Since Ricci flow preserves the local structure of a manifold and the Riemannian metric of the manifold after Ricci flow is uniform (see Section 2), the local relations among the data points across the entire process of the algorithm are preserved. The key concept behind the proposed algorithm is to construct a Ricci flow equation directly on discrete data points without parametrization and meshes. Under the premise of a non-negative curvature, the Ricci flow process transforms the original irregular manifold to a subset of a sphere. Finally, traditional manifold learning algorithms can be used to reduce the dimensionality of the (high dimensional) sphere. It is assumed that the data points are distributed on a single open Riemannian manifold to guarantee that the manifold evolves into a punctured sphere under the Ricci flow. In the current presentation, we only consider Riemannian manifolds with non-negative Ricci curvature. Manifold learning with negative Ricci curvature will be discussed in our next paper. We refer to that style as algorithm-dynamic manifold learning. This paper is just a first attempt in that direction.

1.2 Related work

In recent years, there have been several studies that have studied the intrinsic curvature of sampled points by combining traditional manifold learning techniques in designing new algorithms. Among these, Kwang et al. [15] studied this problem by redefining the group Laplacian operator on the data set. The redefined group Laplacian matrix was used to measure the pair-wise similarities between the data points related to the point to point distance as well as to the curvature information for each local patch. Their experimental results showed that the redefined Laplacian operator led to lower error rates in spectral clustering than the traditional Laplacian operator. Xu et al. [16–18] used the Ricci flow to rectify the pair-wise non-Euclidean dissimilarities among data points. However, these methods did not consider the relation among the different edges. Another nonlinear dimension reduction algorithm is the two dimensional discrete surface Ricci flow [19], which is mainly applied to the three dimensional data points distributed on a two dimensional manifold. Gu et al. [19,20] constructed discrete triangle meshes on three dimensional data points, as well as discrete circle packing metric on the triangle meshes. They proposed the discrete surface Ricci flow algorithm using the Newton method, which can map any curved surface to a normal two-dimensional surface (sphere, hyperbolic space, or plane). Traditional manifold learning algorithms mainly used to reduce the dimension of especially high dimensional data points, where it is difficult to construct the polygon meshes. Note that all these studies which examine applying the Ricci flow to manifold learning are 2-dimensional, no matter whether they are discrete or continuous. They cannot be directly extended to higher dimensional cases.

2 Basic knowledge

If we are given the data set $\{x_1, x_2, \ldots, x_N \in \mathbb{R}^D\}$, where N is the number of data points and D is their dimension, one fundamental assumption of traditional manifold learning is that $\{x_1, x_2, \ldots, x_N\}$ lies on a d-dimensional Riemannian manifold $(\mathcal{M}, g)$ which is embedded in the high dimensional Euclidean space $\mathbb{R}^D (d \ll D)$, where $\mathcal{M}$ is the manifold itself, g is its Riemannian metric defined as the family of all inner products defined on all tangent spaces of $\mathcal{M}$, and $\mathbb{R}^D$ is called the ambient space of $(\mathcal{M}, g)$. On each $T_p\mathcal{M}$, the tangent space at point p, the Riemannian metric is a Euclidean inner product g_p. All of the geometric intrinsic quantities (length, angle, area, volume, and Riemannian curvature) of the Riemannian manifold can be computed under the Riemannian metric g.

2.1 Riemannian curvature

In general, the Riemannian curvature tensor of a d-dimensional ($d \geqslant 3$) Riemannian manifold is represented by a fourth-order tensor, which measures the curvedness of $(\mathcal{M}, g)$ with respect to its ambient space $\mathbb{R}^D$. The directional derivative defined on a Riemannian manifold is a Riemannian connection, represented by ∇. The Riemannian curvature tensor is a $(1, 3)$-tensor defined by

$$\mathcal{R}m(X, Y)Z = \nabla_X \nabla_Y Z - \nabla_Y \nabla_X Z - \nabla_{[X,Y]} Z, \tag{1}$$

on vector fields X, Y, Z [21]. Using the Riemannian metric g, the Riemannian curvature tensor can be transformed into a $(0, 4)$-tensor shown as follows:

$$\mathcal{R}m(X, Y, Z, W) = g(\mathcal{R}m(X, Y)Z, W). \tag{2}$$

The trace of the Riemannian curvature tensor is a symmetric $(0, 2)$-tensor called Ricci curvature tensor [21]:

$$\mathrm{Ric}(Y, Z) = \mathrm{tr}(\mathcal{R}m(\cdot, X)Y). \tag{3}$$

In differential geometry, the Riemannian metric is expressed by the first fundamental form. With respect to the Riemannian sub-manifold, the Riemannian curvature tensor is captured by the second fundamental form $\mathcal{B}(X, Y)$ which is a bilinear and symmetric form defined on tangent vector fields X, Y.

$\mathcal{B}(X,Y)$ is used to measure the difference between the intrinsic Riemannian connection ∇ on $\mathcal{M}$ and the ambient Riemannian manifold connection $\widetilde{\nabla}$ on $\widetilde{\mathcal{M}}$, where $\mathcal{M}$ is embedded into $\widetilde{\mathcal{M}}$. The relation between $\widetilde{\nabla}$ and ∇ is shown by the following Gauss formula [22]:

$$\widetilde{\nabla}_X Y = \nabla_X Y + \mathcal{B}(X,Y). \tag{4}$$

The corresponding relationship between $\mathcal{R}m(X,Y,Z,W)$ of $\mathcal{M}$ and $\widetilde{\mathcal{R}m}(X,Y,Z,W)$ of $\widetilde{\mathcal{M}}$ is shown by the following Gauss equation [22]:

$$\widetilde{\mathcal{R}m}(X,Y,Z,W) = \mathcal{R}m(X,Y,Z,W) - \langle \mathcal{B}(X,W), \mathcal{B}(Y,Z)\rangle + \langle \mathcal{B}(X,Z), \mathcal{B}(Y,W)\rangle. \tag{5}$$

If the ambient space $\widetilde{\mathcal{M}}$ is an Euclidean space $\mathbb{R}^D$ then $\widetilde{\mathcal{R}m}(X,Y,Z,W) = 0$. The Riemannian curvature of $\mathcal{M}$ can be fully captured by the second fundamental form:

$$\mathcal{R}m(X,Y,Z,W) = \langle \mathcal{B}(X,W), \mathcal{B}(Y,Z)\rangle - \langle \mathcal{B}(X,Z), \mathcal{B}(Y,W)\rangle. \tag{6}$$

Under local coordinate system, the second fundamental form $\mathcal{B}$ can be represented by [22]

$$\mathcal{B}(X,Y) = \sum_{\alpha=d+1}^{D} h^{\alpha}(X,Y)\xi_{\alpha}, \tag{7}$$

where ξ_{α} $(\alpha = d+1,\ldots,D)$ is the normal vector field of $\mathcal{M}$ and $h^{\alpha}(X,Y)$ is shown by the second derivative of the embedding map.

2.2 Ricci flow

The Ricci flow is an intrinsic curvature flow on a Riemannian manifold, which is the negative gradient flow of Ricci energy. The Ricci flow is defined by the following geometric evolution time dependent partial differential equation [23]: $\frac{\partial g_{ij}}{\partial t} = -2\mathrm{Ric}_{ij}$, where $g_{ij} = g(\partial_i, \partial_j)$. The Ricci curvature $\mathrm{Ric}(g)$ can be considered as a Laplacian of the metric g, making the Ricci flow equation a variation of the usual heat equation. A solution of a Ricci flow is a one-parameter family of metrics $g(t)$ on a smooth manifold $\mathcal{M}$, defined in a time interval $I \subset \mathbb{R}_+$. In the time interval $I \subset \mathbb{R}_+$, the Riemannian metric $g(t)$ satisfies the metric equivalence condition $e^{-2Ct}g(0) \leqslant g(t) \leqslant e^{2Ct}g(0)$ [23], where $|\mathrm{Ric}| \leqslant C$ and $t \in I$. So the relative geodesic distance between two arbitrary neighborhood points on $\mathcal{M}$ is consistent under the Ricci flow. In general, the solution of the Ricci flow only exists for a short period of time, until the emergence of singular points. In [24], researchers have worked out that the Riemannian manifold of dimension $d \geqslant 4$ can be transformed into a sphere under Ricci flow, when the sectional curvature K satisfies $\frac{\max K}{\min K} < 4$ everywhere.

2.3 Statement of the spherical conditions

Returning to our problem, we note that the patches constructed below on each data point x_i can be either elliptic with positive sectional curvature or hyperbolic with negative sectional curvature. In this paper we only consider the former case and use elliptic polynomial functions to evaluate the local patches on every point, where the detail will be introduced later in this paper. The curvature operator at every point of the Riemannian manifold $\mathcal{M}$ is non-negative. In Ricci flow theory, when the intrinsic dimension $d \leqslant 3$ of $\mathcal{M}$, the Riemannian manifold $\mathcal{M}$ with non-negative Ricci curvature can flow to a sphere under Ricci flow. For $d \geqslant 4$, when the sectional curvature K satisfies $\frac{\max K}{\min K} < 4$ everywhere, the Riemannian manifold $\mathcal{M}$ with positive sectional curvature can flow to a sphere under Ricci flow. So, in the Ricci flow step of the proposed algorithm, we apply the Ricci flow process under the previously mentioned spherical conditions.

3 Algorithm

In practice, it is difficult to analyze the global structure of a nonlinear manifold, especially when there is no observable explicit structure. Our key concept is that we first uncover the local structure of the embedded manifold and then the global structure can be obtained by the alignment of the local structures. Keeping this in mind, we decompose the embedded manifold into a set of overlapping patches and apply the Ricci flow to these overlapping patches independent of each other to avoid singular points, since in general the Ricci flow on a global manifold may encounter singular points when scaling. The global structure of the deformed manifold under the Ricci flow can be obtained from the deformed local patches with a suitable alignment.

3.1 RF-ML algorithm

In this subsection, we describe the algorithm of the RF-ML process. Then, we give a detailed analysis of the steps in this algorithm.

The proposed algorithm is essentially divided into five steps.

(1) Find a local patch (neighborhood) U_i, $i = 1, 2, \ldots, N$ for each data point x_i, $i = 1, 2, \ldots, N$ using the K-nearest neighbor method. To find the K-nearest neighbors, we need to define a distance metric to measure the proximity of arbitrary two points. We use the Euclidean metric as the distance measure in this step.

(2) Construct a special local coordinate system on every point x_i. In the neighborhood U_i, we estimate the local patch information of x_i with a covariance matrix C_i, $C_i = \sum_{x_k \in U_i} (x_k - \overline{x}_i)^{\mathrm{T}} (x_k - \overline{x}_i)$, where $\overline{x}_i$ is the mean vector of the K-nearest data points. The first d eigenvectors $(e_1, e_2, \ldots, e_d)$ with maximal eigenvalues of C_i form a local orthonormal coordinate system of $T_{x_i}\mathcal{M}$. The last $D - d$ eigenvectors $(e_{d+1}, \ldots, e_D)$ form a local orthonormal coordinate system of normal space.

(3) Determine the intrinsic dimension d of local patches by computing the value of the 95% principle components. In practice, owing to the varying curvature of different local patches, the local dimension d_i of each patch U_i may be very inconsistent. We choose $d = \max\{d_1, d_2, \ldots, d_N\}$. If $d = D$, stop the algorithm. The manifold's dimension is not reducible. If $d < D$, continue the algorithm.

(4) Construct Ricci flow equations on local patches and then let the overlapping patches U_i, $i = 1, \ldots, N$ flow independently into local spherical patches Y_i, $i = 1, 2, \ldots, N$ with constant positive Ricci curvature C. Details will be shown below.

(5) Align the discrete spherical patches Y_i, $i = 1, 2, \ldots, N$ into a global subset P of a sphere with positive curvature C, where $P \subset \mathbb{R}^D$. $p_i \in P$ is the corresponding representation of x_i. Details will also be shown below.

(6) Reduce the dimensionality of the subset P using traditional manifold learning algorithms, where the distance metric between two arbitrary points on P is the metric of the sphere other than the Euclidean. The d-dimensional representations are $\{z_1, z_2, \ldots, z_N\} \in \mathbb{R}^d$.

The intuitive representation of the RF-ML algorithm is shown in Figure 1 and the brief algorithm procedure is shown in Algorithm 1.

Algorithm 1 Applying Ricci flow to manifold learning (RF-ML)

Input: Training data points $\{x_1, x_2, \ldots, x_N\} \in \mathbb{R}^D$, neighbor size parameter K.
1. **for** $i = 1$ **to** N **do**
2. Find K-nearest neighbors of x_i;
3. Compute tangent space $T_{x_i}\mathcal{M}$;
4. **end for**
5. **Repeat**
6. Update the Ricci flow equations using Eqs. (11)–(13);
7. **Until** convergence;
8. Align the deformed $\{Y_1, Y_2, \ldots, Y_N\}$ to a complete subset P of sphere;
9. Use traditional manifold learning algorithms to reduce the dimension of spherical data points.

Output: Low dimensional representations $\{z_1, z_2, \ldots z_N\} \in \mathbb{R}^d$.

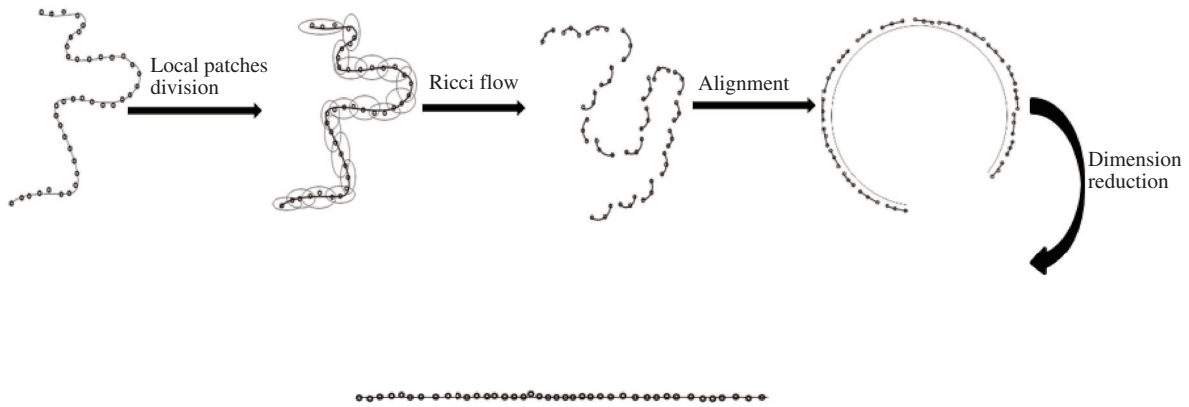

Figure 1 The intuitive process of the RF-ML algorithm. The sub-images from left to right are input data points distributed on manifold $\mathcal{M}$, overlapping patches, overlapping patches flowing into discrete spherical patches using Ricci flow, and alignment of spherical patches to a subset of a sphere, respectively. On the bottom is the representation of data points in a low dimensional Euclidean space.

Step (4) above is to minimize the value of the energy function such that the Ricci curvature at every point converges to a constant curvature. The curvature energy function is shown as follows:

$$E(\mathrm{Ric}) = \int \|\mathrm{Ric} - C\|^2 \mathrm{d}\mathcal{M} \approx \sum_{i=1}^{N} |\mathrm{Ric}(x_i) - C|^2, \tag{8}$$

where C is the value of the constant non-negative Ricci curvature, $\mathcal{M}$ is the embedded manifold, and Ric is called the Ricci curvature. The convergence of the Ricci flow, which we prove in Theorem 1 below, indicates that the energy function can arrive at the optimal solution. To obtain the minimum solution of the curvature energy function, we calculate the step by step solution with the help of the Ricci flow. Referring to the purpose of this paper, it is not in our interest to learn the structure of a single Riemannian manifold $(\mathcal{M}, g)$, but rather to discover a one-parameter family of such manifolds $t \to (\mathcal{M}(t), g(t))$, parameterized by a time parameter t. The dynamic process is controlled by the Ricci flow, until $(\mathcal{M}(t), g(t))$ converges to a Riemannian manifold (H, g) with constant curvature, where the global set of patches P (step (5)) are contained. In Theorem 2, we prove that the Riemannian manifold H is diffeomorphic to the original Riemannian manifold $\mathcal{M}$.

The discrete Ricci flow equation as well as the corresponding iterative equations defined on the discrete data points $\{x_1, x_2, \ldots, x_N\} \in \mathbb{R}^D$ is constructed in the following manner.

Assume that the local coordinates of $x_j \in U_i$ under local ambient coordinates $\langle x_i; e_1, e_2, \ldots, e_d, \ldots, e_D\rangle$ are represented as $(x_j^1, x_j^2, \ldots, x_j^d, \ldots, x_j^D)$. The first d coordinates are seen as the local natural parameters of U_i. A smooth representation of the local patch U_i under the local coordinate system is described by

$$f(x^1, x^2, \ldots, x^d) = (x^1, \ldots, x^d, f^{d+1}(x^1, x^2, \ldots, x^d), \ldots, f^D(x^1, x^2, \ldots, x^d)), \tag{9}$$

where $(x^1, \ldots, x^d)$ is a coordinate chart at U_i.

In this paper, we use the least squares method to approximate the analytic functions f^α, $\alpha = d+1, \ldots, D$ under the local coordinate system constructed above. To guarantee that the curvature operator on each patch satisfies the spherical conditions, as presented in Section 2, we choose second-order elliptic polynomial functions to approximate the structures of local patches. The polynomial form of f^α is depicted as $f^\alpha(x) = W_\alpha \Phi^{\mathrm{T}}$, $\alpha = d+1, \ldots, D$, where $W_\alpha = [a_0^\alpha, a_1^\alpha, \ldots, a_d^\alpha, a_{11}^\alpha, a_{12}^\alpha, \ldots, a_{(dd)}^\alpha]$ is the coefficient vector to be determined by the least squares method and $\Phi = [1, x^1, \ldots, x^d, x^1x^1, x^1x^2, \ldots, x^dx^d]$ is the second-order polynomial basis vector.

Under the local smooth representation f of U_i, the corresponding d tangent vector basis at x_i is given by $\{\frac{\partial f}{\partial x^1}(x_i), \frac{\partial f}{\partial x^2}(x_i), \ldots, \frac{\partial f}{\partial x^d}(x_i)\}$, where $\frac{\partial f}{\partial x^j} = (0, \ldots, 1, \ldots, 0, W_{d+1}\frac{\partial \Phi^{\mathrm{T}}}{\partial x^j}(x_i), \ldots, W_D\frac{\partial \Phi^{\mathrm{T}}}{\partial x^j}(x_i))$. Then the local Riemannian metric tensor is shown as $G_i = [g_{jk}]$, $g_{jk} = \langle \frac{\partial f}{\partial x^j}(x_i), \frac{\partial f}{\partial x^k}(x_i)\rangle$. The second fundamental form coefficient is shown as $h_{jk}^\alpha \doteq \frac{\partial^2 f^\alpha}{\partial x^j \partial x^k}$.

So the local Ricci flow equation defined on x_i is represented as follows:

$$\frac{\partial}{\partial t}\left(\frac{\partial f}{\partial x^j}\cdot\left(\frac{\partial f}{\partial x^k}\right)^{\mathrm{T}}\right)_{x_i}=-2\sum_{\alpha=d+1}^{D}\sum_{l=1}^{d}\left(\frac{\partial^2 f^\alpha}{\partial x^l\partial x^l}\cdot\frac{\partial^2 f^\alpha}{\partial x^j\partial x^k}-\frac{\partial^2 f^\alpha}{\partial x^l\partial x^j}\cdot\frac{\partial^2 f^\alpha}{\partial x^l\partial x^k}\right)_{x_i}. \tag{10}$$

To solve the Ricci flow equation, we need to discretize the differential operators on point clouds. Suppose $V_j=(\frac{\partial f^{d+1}}{\partial x^j},\frac{\partial f^{d+2}}{\partial x^j},\ldots,\frac{\partial f^{D}}{\partial x^j})$, Eq. (10) can be represented as the function of V_j, that is $\frac{\partial}{\partial t}V_j=F(V_j,\nabla V_j)$, where $F(V_j,\nabla V_j)=-\sum_{\alpha=d+1}^{D}\sum_{l=1}^{d}(\frac{\partial^2 f^\alpha}{\partial x^l\partial x^l}\cdot\frac{\partial^2 f^\alpha}{\partial x^j\partial x^j}-\frac{\partial^2 f^\alpha}{\partial x^l\partial x^j}\cdot\frac{\partial^2 f\alpha}{\partial x^l\partial x^j})\cdot(\frac{\partial f}{\partial x^j})^{\mathrm{T}^+}$. Therefore, the local Riemannian metric tensor as well as the corresponding Ricci curvature are iterated under the Ricci flow in the local neighborhood U_i as follows:

$$V_j^{n+1}=V_j^n+\Delta tF(V_j^n,\nabla V_j^n),\quad j=1,\ldots,d, \tag{11}$$

$$g_{jk}^{n+1}=\delta_{jk}+V_j^{n+1}\cdot {V_k^{n+1}}^{\mathrm{T}}, \tag{12}$$

$$\mathcal{R}m_{jk}^{n+1}=\sum_l\left(\nabla_l V_l^{n+1}\cdot\nabla_k {V_j^{n+1}}^{\mathrm{T}}-\nabla_l V_j^{n+1}\cdot\nabla_l {V_k^{n+1}}^{\mathrm{T}}\right). \tag{13}$$

These equations are optimized by updating them until $\mathcal{R}m_{jk}^{T}\to C$, where C is a non-negative constant and T is the number of total iterations.

In step (5) above, after the Ricci flow converges at each x_i, the overlapping local patches $\{U_1,U_2,\ldots,U_N\}$ flow to a set of discrete local spherical patches $\{Y_1,Y_2,\ldots,Y_N\}$. We denote the global coordinates of $\{Y_1,Y_2,\ldots,Y_N\}$ as $\{P_1,P_2,\ldots,P_N\}$, which are related based on a linkage by a set of global alignment maps which shift the discrete local spherical patches to a global subset of a sphere. We propose the global coordinates $p_{i_j}\in P_i$ to satisfy the following equations, such that p_{i_j} is obtained by the local affine transformation of y_{i_j}, where $y_{i_j}\in Y_i$. We have

$$p_{i_j}=\bar{p}_i+Q_i y_{i_j}+\epsilon_j^{(i)},\quad i=1,\ldots,N,\quad j=1,\ldots,K, \tag{14}$$

where N is the number of sample points, K is the nearest neighbor size parameter, Q_i is the unit of orthogonal transformation, and $\epsilon_j^{(i)}$ is the reconstruction error.

To obtain the optimized local affine transformation, we need to minimize the local reconstruction error matrix E_i:

$$\min E_i=\sum_{i=1}^{N}\left\|P_i\left(I-\frac{1}{K}ee^{\mathrm{T}}\right)-Q_iY_i\right\|_F^2, \tag{15}$$

where $E_i=[\epsilon_1^{(i)},\epsilon_2^{(i)},\ldots,\epsilon_K^{(i)}]$.

Minimizing the above least square error is equal to solve the following eigenvalue problem:

$$B=SWW^{\mathrm{T}}S^{\mathrm{T}}, \tag{16}$$

where $S=[S_1,S_2,\ldots,S_N]$, $PS_i=P_i$, $W=\mathrm{diag}(W_{11},W_{22},\ldots,W_{NN})$, $W_{ii}=(I-\frac{1}{K}ee^{\mathrm{T}})(I-Y_i^{\dagger}Y_i)$ and $Y_i^{\dagger}$ the generalized inverse matrix of Y_i.

Decomposing matrix B using the singular value decomposition (SVD) method, we obtain $B=U\Lambda U^{-1}$, where the columns of U are the unit orthogonal eigenvectors of B. Λ is a diagonal matrix and the diagonal components, which are arranged in ascending order, are the eigenvalues of B. The optimal solution is obtained by the D eigenvectors of the matrix B, corresponding to the 2nd to $(D+1)$st smallest eigenvalues of B. However, the optimal data set P that we finally need is distributed on a sphere with curvature C. So we need to give a set of constraints:

$$P_iP_i^{\mathrm{T}}=\frac{1}{C},\quad i=1,2,\ldots,N. \tag{17}$$

Under these constraints, we rewrite matrix B as $B=QRR^{\mathrm{T}}Q^{\mathrm{T}}$, where R is a diagonal matrix, $R_{ii}=\frac{\sqrt[D]{C}}{\sqrt[2]{\|(U_{i1},U_{i2},\ldots,U_{iD})\|_2}}$ $(i=2,\ldots,D+1)$, the rest $R_{jj}=1$ $(j\neq 2,\ldots,D+1)$. The values of the obtained from the 2nd to $(D+1)$st columns of Q are the optimal data set P, which is distributed on a sphere with curvature C.

3.2 Theoretical analysis

In this subsection, we present a theoretical analysis of our algorithm. First we want to illustrate the convergence of the Ricci flow, which has previously been proven by researchers. Second, we derive the relation between the original and the deformed manifolds under the Ricci flow.

Theorem 1 ([25]). Assume that $(\mathcal{M}, g_0)$ has weakly 1/4-pinched sectional curvatures in the sense that $0 \leqslant K(\pi_1) \leqslant 4K(\pi_2)$ for all two-planes $\pi_1, \pi_2 \subset T_p\mathcal{M}$. Moreover, we assume that $(\mathcal{M}, g_0)$ is not locally symmetric. Then the normalized Ricci flow with initial metric g_0 exists for all time, and converges to a constant curvature metric as $t \to \infty$.

Theorem 1 has been proven by Brendle and Schoen [25] in 2008. Generally, the Ricci flow exists for all time and it converges to a constant curvature. For each local patch, the Ricci flow definitely exists and also converges to a positive constant curvature C.

Theorem 2. The global set of patches P (step (5)) is distributed on a subspace H of a sphere, where H is a Riemannian manifold. Additionally, H is diffeomorphic to the original Riemannian manifold M.

Proof. The original data points $\{x_1, x_2, \ldots, x_N\}$ are distributed on M. At each data point x_i, there is a local neighborhood U_i, such that $\sum_i U_i$ can fully cover the manifold M. Thus, M has an enumerable set as a basis.

According to the local Ricci flow, at each data point x_i, there is a diffeomorphism f_i between U_i and Y_i. In the global alignment, there is a diffeomorphism g_i between Y_i and P_i. Thus, the map $g_i \circ f_i$ between U_i and P_i is also a local diffeomorphism. Let

$$l_i(p) = \begin{cases} 1, & p \in U_i, \\ 0, & p \notin U_i, \end{cases} \tag{18}$$

$$o_i = \frac{l_i}{\sum_j l_j}, \quad i = 1, 2, \ldots, N. \tag{19}$$

Define

$$o_i \cdot g_i \circ f_i = \begin{cases} o_i \cdot g_i \circ f_i(p), & p \in U_i, \\ 0, & p \notin U_i, \end{cases} \quad i = 1, 2, \ldots, N. \tag{20}$$

Let $f = \sum_i o_i \cdot g_i \circ f_i$. It is defined on the global manifold $\mathcal{M}$ and induces a local diffeomorphism in any local patch. Since $\mathcal{M}$ has an enumerable set as a basis, according to the unit decomposition theorem of manifolds, f is a global diffeomorphism. Therefore, $f(\mathcal{M})$ is also a Riemannian manifold. It is obvious that the data set P is distributed on the manifold $f(\mathcal{M})$, which is just the manifold H mentioned in the statement of Theorem 2.

4 Experiments

In this section, we compare the proposed RF-ML algorithm with other manifold learning algorithms on several synthetic datasets and on four real world databases.

4.1 Intrinsic dimension

One implicit fundamental assumption of traditional manifold learning is that the dimension of the input data set at each point is consistent. But in practice the distribution of data points on a manifold may not be uniform and the dimensions of the manifold may even differ at the various data points. An intuitive example illustrating this situation is depicted in Table 1. We list the numbers of neighborhoods with different dimensions for six groups of datasets: ORL Face[1)], Yale Face, YaleB Face[2)], Weizmann [26], Swiss Roll [27], and Sphere [27]. We choose the neighborhood-size parameter $K = 10$ and use the PCA algorithm to obtain the 95% principal components. The latter is approximately viewed as the dimension

1) http://www.cs.toronto.edu/~roweis/data/olivettifaces.mat.
2) http://www.face-rec.org/databases/.

Table 1 Number of neighborhoods in each dimension[a)]

Databases	ORL Face	Yale Face	YaleB Face	Weizmann	Swiss Roll	Sphere
Org-dim	1024*	1024*	1024*	200*	3*	3*
1	0	0	0	0	0	0
2	0	0	0	90	1000	984
3	0	0	38	317	–	16
4	0	0	238	366	–	–
5	0	0	362	315	–	–
6	6	0	415	251	–	–
7	214	9	772	349	–	–
8	182	156	583	387	–	–
9	–	–	6	–	–	–
Total	400*	165*	2414*	2075*	1000*	1000*
K	10	10	10	10	10	10
Ratio	0.95	0.95	0.95	0.95	0.95	0.95

a) The first row shows six datasets. The second row shows the original dimension of each dataset. The rows from 3 to 11 list the number of neighborhoods in dimension $1, 2, 3, 4, 5, 6, 7, 8$, and 9, respectively. The "Total" row shows the total number of data in each dataset. "K" denotes the neighborhood-size parameter and "Ratio" denotes the percentage of principal components. Among all the data in Table 1, those labeled by "$*$" are from the original dataset and the other data come from our experiment results.

Table 2 Neighborhood preserving ratio. In this experiment, we fix the neighborhood-size parameter $K = 10$[a)]

Methods	PCA	Isomap	LLE	LEP	LTSA	TLE	RF-ML
Swiss Roll	0.5137	0.8594	0.6187	0.3981	0.6121	**0.8701**	0.8594
Ellipsoid	0.4399	0.6914	0.6205	0.7506	0.4390	0.7246	**0.8702**
Sphere	0.4815	0.6467	0.5213	0.7720	0.5465	0.6871	**0.8684**
Gaussian	0.9969	0.9261	0.9359	0.6406	**0.9970**	0.9421	0.9909

a) The blod fonts represent the highest value of neighborhood preserving ratio.

of the local neighborhoods and may not be the same for all of the neighborhoods. Table 1 shows the number of neighborhoods in each dimension.

4.2 Dimensionality reduction

To evaluate the performance of RF-ML algorithm, we compare the proposed method with several traditional manifold learning algorithms (PCA, Isomap, LLE, LEP, Diffu-Map, LTSA) on four sets of three dimensional data, including: Swiss Roll, Sphere, Ellipsoid and Gaussian. Swiss Roll is a locally flat surface, where the Gauss curvature (i.e., Ricci curvature of two dimensional manifolds) is zero everywhere. However, the Gauss curvatures of the other three datasets are not zero. The objective of this comparison is to map each data set to two dimensional Euclidean space and then to analyze the neighborhood preserving ratios (NPRs) [28] of different algorithms. Table 2 shows the results of our RF-ML compared with seven different algorithms on these four sets of data, where the K-nearest neighbor parameter is set to $K = 10$. The NPR [28] is defined as follows:

$$\mathrm{NPR} = \frac{1}{KN}\sum_{i=1}^{N}\Big|\mathcal{N}(x_i)\bigcap\mathcal{N}(z_i)\Big|. \tag{21}$$

The parameter x_i represents the input data point and z_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 $\mathcal{N}(z_i)$ is the set of $\{l\}$, where z_l is the K-nearest neighbor of z_i. The notation $|\cdot|$ represents the number of intersection points.

NPR measures the local structure preserving ratio of the dimension reduction algorithms. Table 2 shows that for all but the Gaussian dataset, the NPR of RF-ML has the best performance among all of

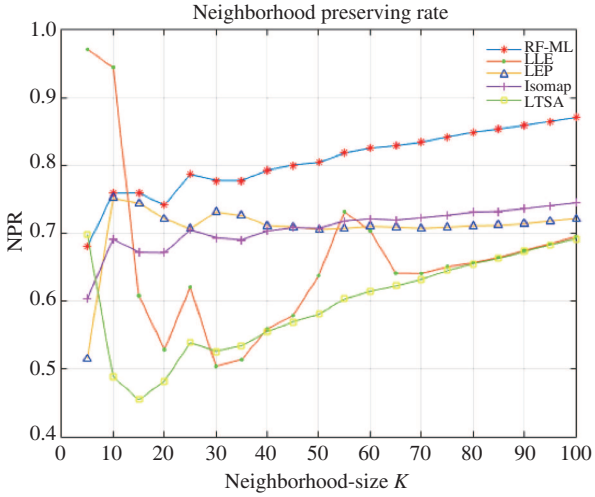

Figure 2 (Color online) The neighborhood preserving rate of two-dimensional Ellipsoid embedded in 3-dimensional Euclidean space. Compute the NPRs under different neighborhood size parameter values K with five manifold learning algorithms.

the algorithms. The Swiss Roll is a locally flat two-dimensional manifold. Its data structure is unchanged under the Ricci flow. Therefore, no Ricci flow is needed. As for the Sphere dataset, its Gauss curvature is a unique constant everywhere. There is no need for the Ricci flow here either. With regard to the Ellipsoid and Gaussian datasets, note that the Gauss curvatures at different points are not always the same. When using RF-ML to reduce their dimensions, the NPR of the Ellipsoid dataset outperforms the other six traditional manifold learning algorithms. For the Gaussian database, owing to the characteristics of the dataset distribution, the NPRs of PCA and LTSA are quite high. In comparison with the six algorithms, the NPR of our method is not bad and is also quite high. Comparing the TLE algorithm with Isomap, we can see that TLE outperforms Isomap in all cases. However, since TLE does not consider the curvature information of datasets, our proposed method achieves better results in the Ellipsoid, Sphere and Gaussian datasets. This clearly demonstrates that RF-ML is more stable and the Ricci flow process is better at preserving the local structure of data points.

Here, we see another advantage of the proposed algorithm. RF-ML is relatively stable under different values of the neighbor size parameter K. Stable in this section means that the change of NPRs under different values of K is both relatively small and smooth, while traditional manifold learning algorithms are more sensitive to the neighbor-size parameter K. That is because those traditional algorithms implicitly assume that the local patch at each point is flat. Our method tries to find the intrinsic curvature structure of the local patch at every point and thus, is not sensitive to the neighbor size parameter K. In this experiment, we evaluated the performance of our RF-ML algorithm compared with the other four algorithms (LLE, LEP, Isomap, and LTSA) under different values of neighbor size parameter K on the Ellipsoid dataset. The comparison results are shown in Figure 2. As the neighborhood-size parameter K values are increasing, the NPR of RF-ML grows continuously. But the NPRs of the other four manifold learning algorithms are unstable and especially sensitive to the different values of neighbor size parameter K.

Finally, to show the intuitive embedding of the synthetic datasets, we give two intuitive examples shown in Figures 3 and 4. These two figures illustrate the intuitive embedding results of the puncture sphere and Gaussian datasets under five different manifold learning algorithms.

4.3 Real world databases

In this subsection, we present the results of our experiments on four real world databases: ORL database, Yale Face database, YaleB Face database, and USPS database[3)].

3) http://www-stat.stanford.edu/~tibs/ElemStatLearn/data.html.

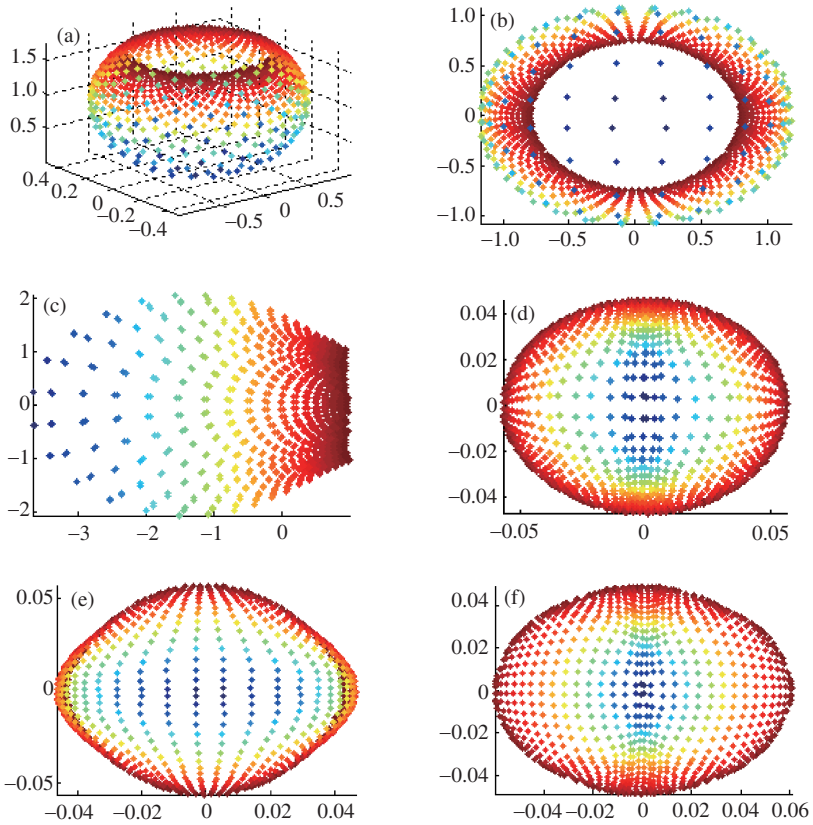

Figure 3 (Color online) The intuitive embedding of puncture sphere database under different algorithms. (a) The original database; (b) Isomap; (c) LLE; (d) LEP; (e) LTSA; (f) RF-ML.

The ORL Face database contains 10 different images of 40 distinct subjects. In total, this database contains 400 images. For some subjects, the images were taken at different times, varying the lighting, facial expressions, and facial details.

The Yale Face database contains 165 grayscale images of 15 individuals. There are 11 images per subject. These consist of one image per different facial expression or configuration: center-light, glasses, happy, left-light, no glasses, normal, right-light, sad, sleepy, surprised, and wink.

The YaleB Face database contains 5760 single images of 10 subjects. There are 576 images per subject. For every subject in a particular pose, an image with ambient illumination was also captured.

The USPS database refers to numeric data obtained from the scanning of handwritten digits from envelopes by the U.S. Postal Service. There are 10 classes of these handwritten digits from 0 to 9. The total number of images in this database is 9298. The size of these images has been normalized, resulting in 16×16 grayscale images. Herein, we choose 700 images for each digital class.

The proposed algorithm aims to regularize the curvature distribution of each database using the Ricci flow. We first analyzed the curvature distributions of these four databases. The intuitive results are shown in Figure 5. From these four histograms in Figure 5, we can see that the curvature distribution of the USPS database is more variable than it is for the other three databases. One reason is that USPS database have enough samples to guarantee that the sample distribution is sufficiently dense. The other reason is that the data points among each classes vary greatly. For the other three databases, the curvature distributions ranged from 0 to 0.01. The total size of the ORL and Yale Face databases are 400 and 165, respectively. The distribution within these databases in high dimensional Euclidean space is especially sparse. Therefore, the curvature on each data point is relatively small.

Next, we compared the proposed algorithms with traditional manifold learning algorithms on these four databases mentioned above to test their classification performance. In this experiment, we used both traditional manifold learning algorithms and RF-ML algorithm to reduce the dimensions of these

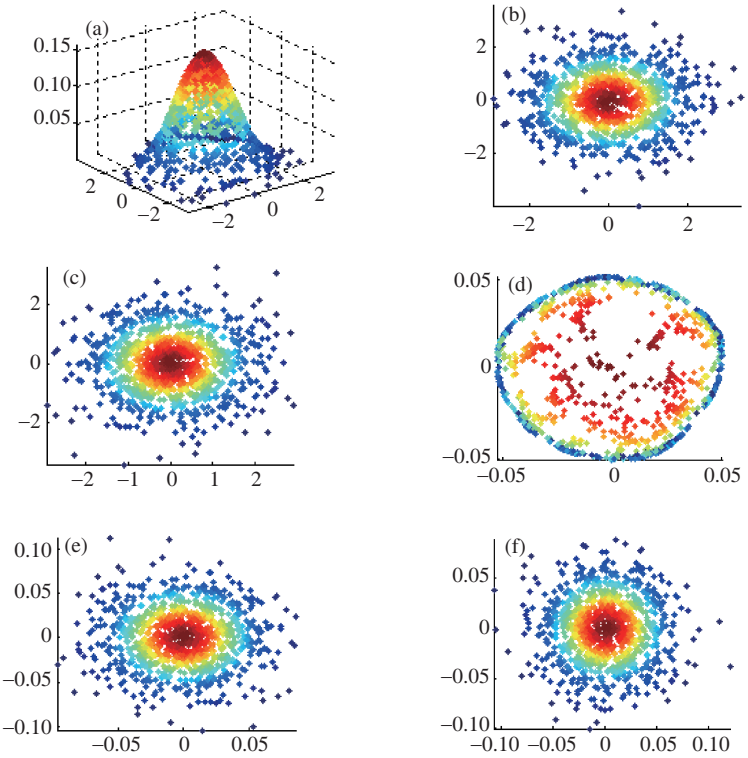

Figure 4 (Color online) The intuitive embedding of Gaussian database under different algorithms. (a) The original database; (b) Isomap; (c) LLE; (d) LEP; (e) LTSA; (f) RF-ML.

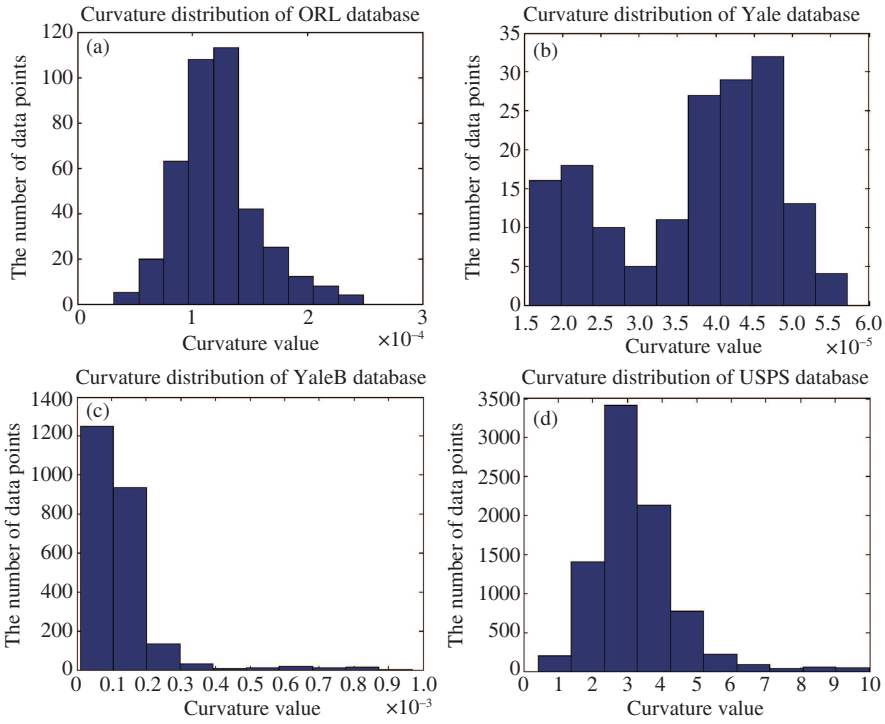

Figure 5 (Color online) The curvature distributions of the (a) ORL face database, (b) Yale face database, (c) YaleB face database, and (d) USPS database.

four databases. Then we use a nearest neighbor classifier method to test the recognition accuracies of the different algorithms on these four databases. In the classification step, for these four databases, we

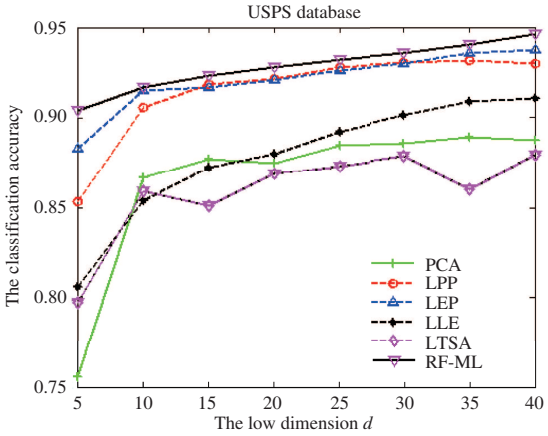

Figure 6 (Color online) The classification accuracies of different manifold learning algorithms under different low dimensions d.

Table 3 Neighborhood preserving ratio[a)]

Method	PCA	LPP	LLE	LEP	LTSA	RF-ML
ORL DB	52.50 ± 1.2	62.50 ± 2.1	61.87 ± 1.6	63.82 ± 2.3	60.31 ± 1.6	65.04 ± 1.3
Yale DB	35.73 ± 1.4	71.24 ± 1.8	69.05 ± 2.2	72.06 ± 1.5	70.43 ± 1.4	73.28 ± 1.6
YaleB DB	63.72 ± 1.7	67.26 ± 1.2	60.46 ± 1.7	72.62 ± 1.8	65.76 ± 1.3	73.95 ± 2.1
USPS DB	86.69 ± 0.9	91.61 ± 1.3	84.62 ± 1.5	92.53 ± 1.1	86.12 ± 1.4	93.01 ± 1.7

a) In this experiment, we fixed the neighborhood size parameter at $K = 10$. In the dimension reduction step, we choose $d = 6, 8, 7, 10$ as the intrinsic dimensions of the ORL DB, Yale DB, YaleB DB, and USPS DB. In the recognition step, we chose half of each dataset as the training dataset and used the rest as the testing dataset.

chose half of the images of each distinct subject as the training subset and the rest of the images of each subject were used as the test subset. The experimental results are shown in Table 3. From Table 3, where we can see that RF-ML mostly outperforms the other five traditional manifold learning algorithms. The proposed method analyzed the curvature information of these four databases and uses Ricci flow to regularize the curvature information. Thus, in comparison with traditional manifold learning algorithms, we uncovered the intrinsic curvature information of these databases and added the information into the dimension reduction process.

Finally, we further compared our RF-ML algorithm with these four algorithms on the USPS database to test the classification accuracies under different low dimension d. In comparison with the other three real world databases, the curvature distribution of the USPS database varied higher than the other databases. In this way, we did run sufficiently comparative experiments on the USPS database to test the performance of these algorithms under different low dimensions d. The final experimental results are shown in Figure 6. From this figure, we can see that the classification accuracies of our algorithm mostly outperform other traditional manifold learning algorithms under different low dimensions d. Overall, using the Ricci flow to regularize the curvature distribution of datasets is a significant improvement of manifold learning.

5 Conclusion and future work

In the field of image recognition where it is necessary to precisely describe the continuously changing images, one critical step is to assume that the image set is distributed on a low dimensional manifold, which is embedded in the high dimension pixel space. Then, it is possible to use mathematical knowledge of manifolds to deal with these datasets using techniques such as dimensionality reduction, classification, clustering, and recognition among others. Manifold learning is an effective way to link the classical geometry with machine learning. Discovery of the precise manifold structure has direct impact the

learning results. Many traditional manifold learning algorithms do not differentiate between the varying curvature at different points manifolds. The proposed aims to excavate the power of the Ricci flow and to use it to dynamically deform the local curvature to make the manifold's curvature uniform. Extensive experiments were conducted, which have shown that the proposed method is more stable in comparison with other traditional manifold learning algorithms.

Several researches based on the application of Ricci flow to manifold learning have been reported. However, to the best of our knowledge, this is the first study to have applied the Ricci flow to high dimensional (unlimited dimension) data points for dimensionality reduction. One limitation of our algorithm is that RF-ML only works for manifolds with non-negative curvature. We will discuss the applicability of the curvature flow algorithm to manifolds with negative Ricci curvature in our next paper. We believe that there will be la significant number of applications to manifold learning where curvature flow is relevant.

Acknowledgements This work was supported by National Key Research and Development Program of China (Grant No. 2016YFB1000902), National Natural Science Foundation of China (Grant Nos. 61472412, 61621003), Beijing Science and Technology Project, and Tsinghua-Tencent-AMSS-Joint Project.

References

1 Roweis S T, Saul L K. Nonlinear dimensionality reduction by locally linear embedding. Science, 2000, 290: 2323–2326
2 Tenenbaum J B, Silva V D, Langford J C. A global geometric framework for nonlinear dimensionality reduction. Science, 2000, 290: 2319–2323
3 Lin B B, He X F, Ye J P. A geometric viewpoint of manifold learning. Appl Inform, 2015, 2: 3
4 Zhu T, Shen F R, Zhao J X, et al. Topology learning embedding: a fast and incremental method for manifold learning. In: Proceedings of International Conference on Neural Information Processing, 2007. 43–52
5 Belkin M, Niyogi P. Laplacian eigenmaps and spectral technique for embedding and clustering. Neural Inform Process Syst, 2001, 14: 585–591
6 He X F, Partha N. Locality preserving projections. In: Proceedings of International Conference Advances in Neural Information Processing Systems, 2003
7 Li Y Y, Lu R Q. Locality preserving projection on SPD matrix Lie group: algorithm and analysis. Sci China Inf Sci, 2018, 61: 092104
8 Zhang Z Y, Zha H B. Principal manifolds and nonlinear dimension reduction via local tangent space alignment. SIAM J Sci Comput, 2005, 26: 313–338
9 Donoho D L, Grimes C. Hessian eigenmaps: locally linear embedding techniques for high-dimensional data. Proc Natl Acad Sci USA, 2003, 100: 5591–5596
10 Cox T F, Cox M A A. Multidimensional Scaling. Boca Raton: CRC Press, 1990
11 Jolliffe I T. Principal Component Analysis. Berlin: Springer, 1989
12 Hettiarachchi R, Peters J F. Multi-manifold LLE learning in pattern recognition. Pattern Recogn, 2015, 48: 2947–2960
13 Mahapatra S, Chandola V. S-Isomap++: multi manifold learning from streaming data. In: Proceedings of IEEE International Conference on Big Data, 2017
14 Cao H D, Zhu X P. A complete proof of the Poincaré and geometrization conjectures-application of the Hamilton-Perelman theory of the Ricci flow. Asian J Math, 2006, 10: 165–492
15 Kwang I K, James T, Christian T. Curvature-aware regularization on Riemannian submanifolds. In: Proceedings of IEEE International Conference on Computer Vision, 2013
16 Xu W P, Hancock E R, Wilson R C. Ricci flow embedding for rectifying non-Euclidean dissimilarity data. Pattern Recogn, 2014, 47: 3709–3725
17 Xu W P, Hancock E R, Wilson R C. Regularising the Ricci flow embedding. In: Proceedings of Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR), 2010. 579–588
18 Xu W P, Hancock E R, Wilson R C. Rectifying non-Euclidean similarity data using Ricci flow embedding. In: Proceedings of the 20th International Conference on Pattern Recognition, 2010
19 Zeng W, Dimitris S, Gu D. Ricci flow for 3D shape analysis. IEEE Trans Pattern Anal Mach Intell, 2010, 32: 662–677
20 Zeng W, Yin X T, Zeng Y, et al. 3D face matching and registration based on hyperbolic Ricci flow. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2008
21 Peter P. Riemannian Geometry. Berlin: Springer, 1998
22 John M. Riemannian Manifolds. Berlin: Springer, 1997
23 Cao H D, Chow B. Recent developments on the Ricci flow. Bull Amer Math Soc, 1999, 36: 59–75
24 Simon B, Richard S. Riemannian manifolds of positive curvature. In: Proceedings of the International Congress of Mathematicians, 2000
25 Brendle S, Schoen R M. Classification of manifolds with weakly 1/4-pinched curvatures. Acta Math, 2008, 200: 1–13
26 Gorelick L, Blank M, Shechtman E, et al. Actions as space-time shapes. IEEE Trans Pattern Anal Mach Intell, 2007, 29: 2247–2253
27 Wittman T. Mani Matlab demo. 2005. http://www.math.umn.edu/~wittman/mani/
28 Sanguinetti G. Dimensionality reduction of clustered data sets. IEEE Trans Pattern Anal Mach Intell, 2008, 30: 535–540