Front. Electr. Electron. Eng. China 2011, 6(1): 27–42
DOI 10.1007/s11460-011-0124-4

RESEARCH ARTICLE

Xiaofei HE, Binbin LIN

Tangent space learning and generalization

Abstract Manifold learning has attracted considerable attention over the last decade, in which exploring the geometry and topology of the manifold is the central problem. Tangent space is a fundamental tool in discovering the geometry of the manifold. In this paper, we will first review canonical manifold learning techniques and then discuss two fundamental problems in tangent space learning. One is how to estimate the tangent space from random samples, and the other is how to generalize tangent space to ambient space. Previous studies in tangent space learning have mainly focused on how to fit tangent space, and one has to solve a global equation for obtaining the tangent spaces. Unlike these approaches, we introduce a novel method, called persistent tangent space learning (PTSL), which estimates the tangent space at each local neighborhood while ensuring that the tangent spaces vary smoothly on the manifold. Tangent space can be viewed as a point on Grassmann manifold. Inspired from the statistics on Grassmann manifold, we use intrinsic sample total variance to measure the variation of estimated tangent spaces at a single point, and thus, the generalization problem can be solved by estimating the intrinsic sample mean on Grassmann manifold. We validate our methods by various experimental results both on synthetic and real data.

Keywords tangent space learning, machine learning, manifold learning

1 Introduction

In many cases of interest in machine learning and data analysis, the data is usually represented as points in a very high-dimensional space. However, intuitively, the data may be generated by structured systems with much fewer degrees of freedom. Various researchers have considered the case when the data is sampled from a submanifold of an ambient space. Consequently, it is crucial to estimate the intrinsic properties of the manifold from random points. These problems are typically referred to as *manifold learning*.

1.1 Manifold learning and its applications

The central problem in manifold learning is to explore the geometry and topology of the manifold from random sampled data points. Figure 1 gives an overview of the canonical manifold learning techniques. The early works aim to find an optimal Euclidean embedding of the data manifold. Principal component analysis (PCA) [1] considers the case when the data manifold is flat and find the maximum variance directions. In the last decade, there is considerable interest to study the nonlinear structure of the manifold. We summarize these works to three groups: kernel variation of PCA, learning with operator and distance, and tangent space learning.

One natural nonlinear extension of PCA is kernel PCA [2], which performs PCA in the feature space. Interestingly, Ref. [3] shows Isomap [4], locally linear embedding (LLE) [5], and Laplacian eigenmaps (LE) [6] can be viewed as a kernel PCA with specific kernels. Isomap, LLE, and LE are the most popular algorithms in manifold learning. Isomap finds a lower dimensional embedding such that the Euclidean distances in the reduced space can well approximate the geodesic distances in the original manifold. LLE finds an embedding such that each data points can be reconstructed by linear combination of its neighbors using the same coefficients as in the original manifold. LE constructs a nearest neighbor graph and aims to find an embedding that preserves the graph structure. There are many extensions of these algorithms, e.g., C-Isomap [7] and modified locally linear embedding (MLLE) [8]. Inspired by the work of Ref. [3], maximum variance unfolding (MVU) [9] was developed by choosing suitable kernel based on isometry criterion.

Received July 14, 2010; accepted October 7, 2010

Xiaofei HE (✉), Binbin LIN
State Key Lab of CAD & CG, College of Computer Science, Zhejiang University, Hangzhou 310058, China
E-mail: xiaofeihe@cad.zju.edu.cn

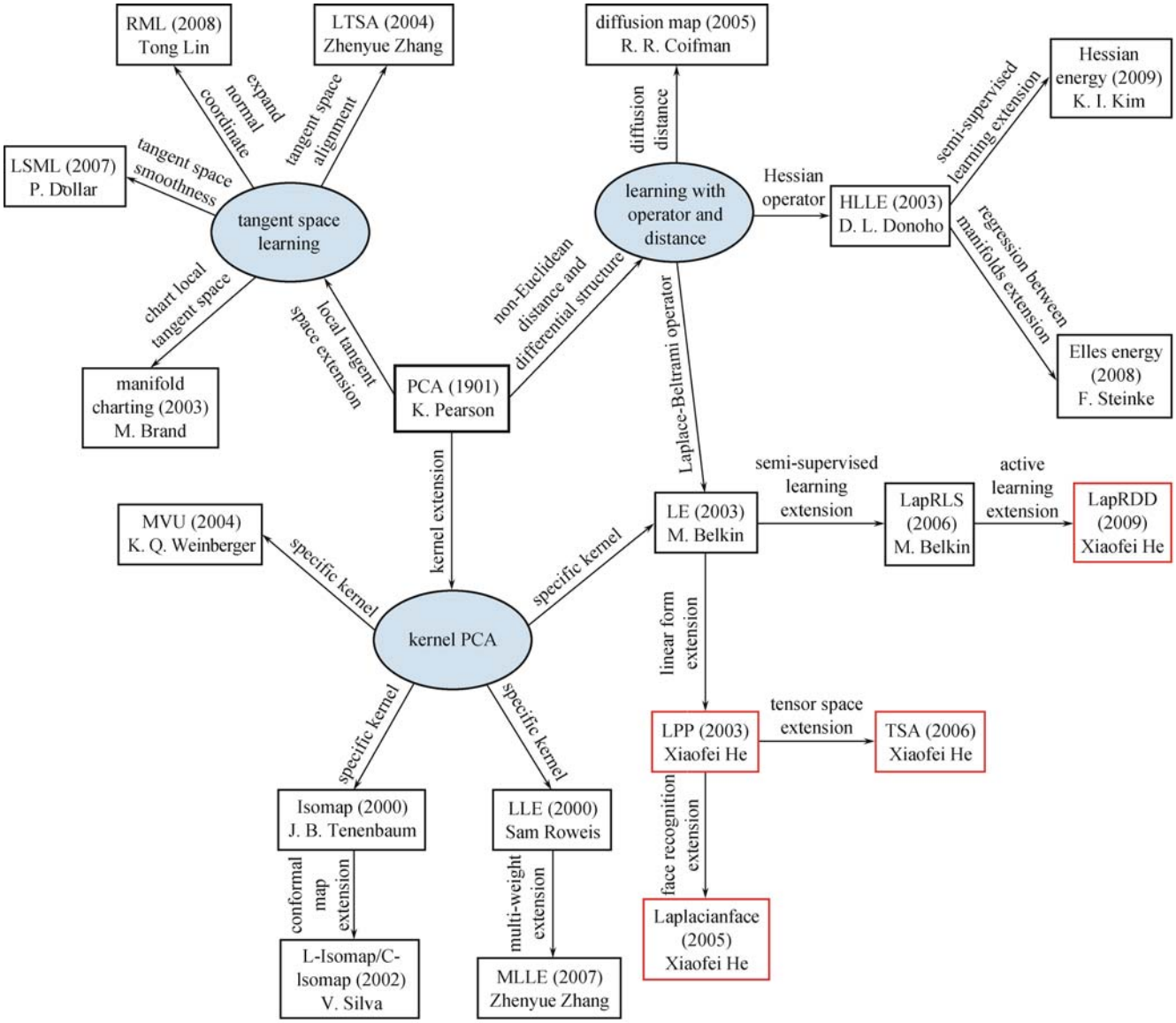


Fig. 1 Overview of manifold learning algorithms

Learning with operator and distance is widely used in machine learning. Traditional methods mainly consider Euclidean distance of data points that ignore the non-Euclidean structure of the manifold. Various non-Euclidean distances are thus developed recently. The representative work is diffusion distance and diffusion map [10].

In manifold learning, two of the most important operators are Laplace-Beltrami operator and Hessian operator. LE discretizes Dirichlet functional, which actually minimize the norm of Laplacian. The discretization of LE was attributable to spectral graph theory [11]. Various theoretical results are developed on LE. Out-of-sample extensions have been proposed in Ref. [12]. The convergence of LE has been proved recently [13].

Laplacian operator was widely used in many other problems of machine learning besides dimensionality reduction, including semi-supervised learning and active learning. In many real world problems, the dimensionality of ambient space can be prohibitively high; thus, there is curse of dimensionality problem in many machine learning tasks, such as semi-supervised learning. One of the natural solutions for this problem is to apply some sort of smoothness criteria. Previous studies have shown that Laplace operator is an ideal operator for measuring the smoothness of the functions on manifold. Therefore, in many semi-supervised learning problems, Laplace operator is used as a regularization term, e.g., Laplacian regularized least squares (LapRLS) [14].

Active learning from manifold perspective also benefits from Laplacian operator [15,16]. In practice, the labels are usually expensive, and the challenge is, thus, how to determine which unlabeled samples would be the most informative (i.e., improve the classifier the most) if they were labeled and used as training samples. He et al. consider the problem of active learning of a regression

model in the context of experimental design [15,16]. Different from traditional approaches whose loss functions are only defined on the labeled points, the proposed algorithm is defined on both measured and unmeasured points.

Locality preserving projection (LPP) [17] is driven from the same criterion of LE, which stands a class of linear manifold learning algorithms. Different from traditional statistical methods like PCA, LPP uses much more local geometrical properties of manifold. The locality preserving property is due to the use of nearest neighbor graph and the graph Laplacian constructed over it. It would be important to note that different graph structures lead to different algorithms, please see Ref. [18] for details. Tensor subspace analysis [19] is the tensor extension of LPP. It considers the case when the data is not in vector space but tensor space.

Besides Laplacian operator, Hessian operator is another very important differential operator. Hessian-based manifold learning methods were proposed recently, including Hessian eigenmaps (HLLE) [20] and Elles energy-based methods [21,22]. HLLE tries to find a faithful embedding by minimizing Hessian energy which is based on isometry criterion. References [21,22] extends the Hessian energy to Elles energy which appears to be suitable for regression and semi-supervised learning.

Tangent space learning is another important direction of manifold learning. Local tangent space alignment (LTSA) [23] tries to construct the global coordinate via local tangent space alignment. Manifold charting [24] has a similar strategy that tries to expand the manifold by splicing local charts. Riemannian manifold learning (RML) [25] uses normal coordinate to expand the manifold, which preserves the metric of the manifold. The embedding quality of these methods mostly depends on the estimation of tangent spaces. Locally smooth manifold learning (LSML) [26,27] tries to learn smooth tangent spaces by adding a regularization term.

1.2 Geometry and topology

Manifold learning can be restated in the following more general framework. Given data points $X = \{x_1, x_2, \ldots, x_n\} \subset \mathcal{M} \subset \mathbb{R}^m$, where $\dim \mathcal{M} = d$ and $d \ll m$. We are trying to learn a function between two manifolds, $f : \mathcal{M} \to \mathcal{N}$. There are many criteria for defining the optimal mapping, and each one can be formulated as a functional minimization problem:

$$\min E(f) = \int_{\mathcal{M}} L(f),$$

where L is a certain operator. In discrete case, we minimize

$$\min \sum_i L(f)_{x_i}.$$

Traditional manifold learning considers the case that the target manifold is a Euclidean space, i.e., $\mathcal{N} = \mathbb{R}^d$. This is the most convenient case since we can compute geodesic distance very easily in Euclidean space. However, such convenience may incur trouble when $\mathcal{M}$ is not isometric to Euclidean space. Even worse, if the topology of $\mathcal{M}$ is not trivial, which means it has holes, there is no one-to-one mapping. Thus, there will be loss of information, and even loss of meaningful structure of the manifold. Consider the case when $\mathcal{M}$ is sphere and $\mathcal{N}$ is a plane, there is no one-to-one mapping, let alone isometric mapping.

Thus, the first problem we should study is the topology of $\mathcal{M}$, and then, it is possible for us to judge whether there is a good f. Computing the topology of $\mathcal{M}$ via random sample points is rather difficult. Data points are discrete, while the topology is a continuum concept. Therefore, we should first construct a continuum object to approximate the manifold. There are two famous work on this topic. One is *persistent homology* [28], and the other is *computing the homology with high confidence* [29]. Both of them first construct simplicial complex via data points. The construction is determined by the radius of ball centered at each data point. If the radius is too small, the balls will be disconnected, leading to incorrect simplicial complex. If it is too large, it will cover the intrinsic hole. For the second work, there is strong requirement for sampling. Persistent homology uses another idea to judge whether the constructed simplicial complex is good. Specifically, it allows the radius to gradually change and finds a persistent period such that the homology is stable.

From the geometrical perspective, the design of $E(f)$ reflects the geometrical properties one tries to preserve. We first consider the isometry criterion. Let $\mathrm{d}f$ be the differential of f. For each x, $\mathrm{d}f_x$ is a linear mapping from the tangent space $T_x\mathcal{M}$ to tangent space $T_{f(x)}\mathcal{N}$. If $\mathrm{d}f$ is orthogonal, then f is an isometry. Let λ_j be the eigenvalue of $\mathrm{d}f$, then

$$E(f) = \int_{\mathcal{M}} L(f) = \int_{\mathcal{M}} \sum_j (\lambda_j - 1)^2 \qquad (1)$$

is a functional that measures the rigid property of the mapping. As we know, isometry is difficult to achieve, and in many cases it is even impossible. An alternative way is to relax this criterion to finding a harmonic mapping [30], since isometric is also harmonic. For harmonic mappings, the functional $E(f)$ reads $E(f) = \int_{\mathcal{M}} e(f)$, where $e(f)$ is the energy density of f,

$$e(f)_x = \frac{1}{2}\|\mathrm{d}f_x\|^2_{\text{H-S}}, \qquad (2)$$

where $\|\cdot\|_{\text{H-S}}$ denotes the Hilbert-Schmidt norm of $\mathrm{d}f_x$. $E(f)$ is called total energy of f. The Hilbert-Schmidt

norm $\|\mathrm{d}f_x\|_{\text{H-S}}$ of its differential at x is defined by

$$\|\mathrm{d}f_x\|^2 = \sum_i h(\mathrm{d}f_x(e_i), \mathrm{d}f_x(e_i)), \tag{3}$$

where $\{e_i\}$ is an orthonormal basis for $T_x\mathcal{M}$, and h is the Riemannian metric of $\mathcal{N}$. The solution of the above problem is a harmonic map. Similar to LE, the total energy is also a good choice for measuring the smoothness of functions on manifolds. The Elles energy [21] and Hessian energy [22] are recently proposed for regression between manifolds. They consider the following functional:

$$E(f) = \int_{\mathcal{M}} \|\nabla' \mathrm{d}f\|^2.$$

The reason is that if f satisfies $\nabla' \mathrm{d}f = 0$, then f has total geodesic property, which means that it maps geodesic on $\mathcal{M}$ to geodesic on $\mathcal{N}$.

From the discretization point of view, both simplicial complex and tangent spaces describe the high-dimensional properties of the manifold. However, the requirement for constructing simplicial complex is crucial, and it is not so intuitive for estimating the geometry of the manifold. While for tangent spaces, almost all differential operators are defined on it. Thus it is a natural choice for analyzing the geometry of the manifold.

1.3 Tangent space learning

Tangent space is a fundamental tool to study the geometry of the manifold. Most of the differential operators on manifold are defined on tangent spaces. Besides, tangent space also shows the relationship between the manifold and ambient space.

Previous work for tangent space learning mainly focuses on how to fit tangent space by neighborhood points. Non-local manifold tangent learning [31] uses the weighted projection distance to measure the fitness. LSML uses similar strategy by introducing a regularization term. However, both of them highly depend on the choice of neighborhood size. Non-local method also uses distant points; however, it may be biased according to the curvature of the manifold. The advantage of such methods is that the smoothness of tangent spaces can be guaranteed. It would be important to note that both of them need to solve a global optimization problem. Therefore, the estimation of tangent space at a single point would be computationally expensive.

In this paper, we propose a novel tangent space learning algorithm called *persistent tangent space learning* (PTSL). Unlike non-local manifold tangent learning and LSML which are global methods, our algorithm performs at each local neighborhood. Intuitively, when estimating the tangent spaces at different points, the local neighborhood sizes should also be different. Therefore, the basic idea of our algorithm is to adaptively change the neighborhood size until the obtained tangent space reaches a persistent period. We represent the tangent spaces as points in a Grassmann manifold. Using statistics on Grassmann manifold, the persistence is thus measured by the variance of the tangent spaces with different neighborhood size. We also discuss how to generalize the tangent space learning to the ambient space.

2 Geometrical perspective on machine learning

In this section, we provide a brief review of our work on machine learning from geometrical perspective. To be specific, we consider the special case where the data points are sampled from an underlying submanifold embedded in the ambient Euclidean space. Particularly, we consider the following problems:

1) How to find a mapping $f : \mathcal{M}^d \subset \mathbb{R}^n \to \mathbb{R}^d$ such that the geometrical properties can be best preserved?

2) How to find the most informative points on the data manifold so that the learning performance (e.g., the regression model) can be improved the most if these points are used as training points?

The first problem is related to manifold embedding, and the second problem is related to active learning. In the following, we provide a brief description of our work to solve these two problems.

2.1 Locality preserving projection and Laplacianfaces

2.1.1 LPP

LPP [17] is a linear dimensionality reduction algorithm. To preserve locality, it minimizes the following integral on manifold:

$$\arg \min_{\|f\|_{L^2(\mathcal{M})}=1} \int_{\mathcal{M}} \|\nabla f\|^2, \tag{4}$$

which is equivalent to

$$\arg \min_{\|f\|_{L^2(\mathcal{M})}=1} \int_{\mathcal{M}} \mathcal{L}(f) f, \tag{5}$$

where the integral is taken with respect to the standard measure on the Riemannian manifold. $\mathcal{L}$ is the Laplace-Beltrami operator on the manifold, i.e., $\mathcal{L}f = -\mathrm{div}\nabla(f)$. Thus, the optimal f has to be the eigenfunction of $\mathcal{L}$.

In practice, the manifold is unknown. By spectral graph theory [11], the manifold can be faithfully modeled by a nearest neighbor graph G with weight matrix S. We consider a linear function $f(\boldsymbol{x}) = \boldsymbol{a}^{\mathrm{T}}\boldsymbol{x}$. Let $X = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m)$ be the data matrix. Thus, the in-

tegral can be approximated as follows:

$$\int_{\mathcal{M}} \mathcal{L}(f) f \approx \boldsymbol{a}^{\mathrm{T}} X L X^{\mathrm{T}} \boldsymbol{a}, \tag{6}$$

$D = \mathrm{diag}(S\mathbf{1})$, and $L = D - S$ is the so called graph Laplacian [11]. $\mathbf{1}$ is a vector of all ones. Similarly, the norm can be approximated as follows:

$$\|f\|_{L^2(\mathcal{M})} \approx \boldsymbol{a}^{\mathrm{T}} X D X^{\mathrm{T}} \boldsymbol{a}. \tag{7}$$

Therefore, the optimal projection is given by solving the following generalized eigenvector problem:

$$X L X^{\mathrm{T}} \boldsymbol{a} = \lambda X D X^{\mathrm{T}} \boldsymbol{a}. \tag{8}$$

Our previous studies have shown that PCA, linear discriminant analysis, and LPP arise from the same principle applied to the difference choices of the graph structure [18].

LPP is especially suitable for clustering [32]. In Ref. [33], we provide a theoretical analysis on the objective function when the manifold has multiple connected components [33].

Theorem 1 Suppose the m-dimensional manifold $\mathcal{M}$ contains k connected components, $\mathcal{M} = \{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_k\}$, and is embedded in $\mathbb{R}^N$. Denote their affine hulls by $\mathrm{aff}(\mathcal{M}_1)$, $\mathrm{aff}(\mathcal{M}_2)$,..., $\mathrm{aff}(\mathcal{M}_k)$, and the corresponding linear spaces by $\mathrm{lin}(\mathcal{M}_1)$, $\mathrm{lin}(\mathcal{M}_2)$,..., $\mathrm{lin}(\mathcal{M}_k)$. Then, the following statements are equivalent:

1) $\int_{\mathcal{M}} \|\nabla f_{\mathcal{M}}\|^2 = 0$,
2) $f_{\mathrm{aff}(\mathcal{M}_i)} = \mathrm{const}, i = 1, 2, \ldots, k$,
3) $\nabla f \perp \mathrm{lin}(\mathcal{M}_1) \oplus \mathrm{lin}(\mathcal{M}_2) \oplus \cdots \oplus \mathrm{lin}(\mathcal{M}_k)$.

The third statement of this theorem describes the connection between the affine hull of the manifold and the optimal function on the ambient space. In this case, the gradient of the function on ambient space is orthogonal to the affine hull of the connected components of the manifold. In some sense, each connected component with the overlapping affine hull will be *collapsed* by the optimal projection. Usually, data points sampled from different components correspond to different objects, and the optimal projection will separate them very well. Thus, this functional is very suitable for clustering when there are multiple near parallel connected components.

2.1.2 Laplacianfaces

By using LPP, we have proposed a novel face recognition algorithm called *Laplacianfaces* [18]. Consider a simple example of image variability. Imagine that a set of face images are generated while the human face rotates slowly. Intuitively, the set of face images correspond to a continuous curve in image space since there is only one degree of freedom, that is, the angle of rotation. Thus, we can say that the set of face images are intrinsically one-dimensional. In face recognition, the face images are mapped to a subspace in which recognition is performed. The subspace is spanned by the eigenvectors of Eq. (8). We can display the eigenvectors as images that are called Laplacianfaces. Using the YALE face database as the training set, we present the first 10 Laplacianfaces in Fig. 2, together with Eigenfaces [34] and Fisherfaces [35]. The Laplacianfaces are the optimal linear approximations to the eigenfunctions of the Laplace Beltrami operator on the face manifold. In this way, the face manifold structure can be best preserved.

Figure 3 shows an example that the face images with various pose and expression of a person are mapped into two-dimensional (2D) subspace. The face image data set used here is the same as that used in Ref. [5]. This data set contains 1965 face images taken from sequential frames of a small video. The size of each image is 20×28 pixels, with 256 gray-levels per pixel. Thus, each face image is represented by a point in the 560-dimensional ambient space. However, these images are believed to come from a submanifold with few degrees of freedom. We leave out 10 samples for testing, and the remaining 1955 samples are used to learn the Laplacianfaces. As can be seen, the face images are mapped into a 2D space with continuous change in pose and expression. The representative face images are shown in the different parts of the space. The face images are divided into two parts. The left part includes the face images with open mouth, and the right part includes the face images with closed

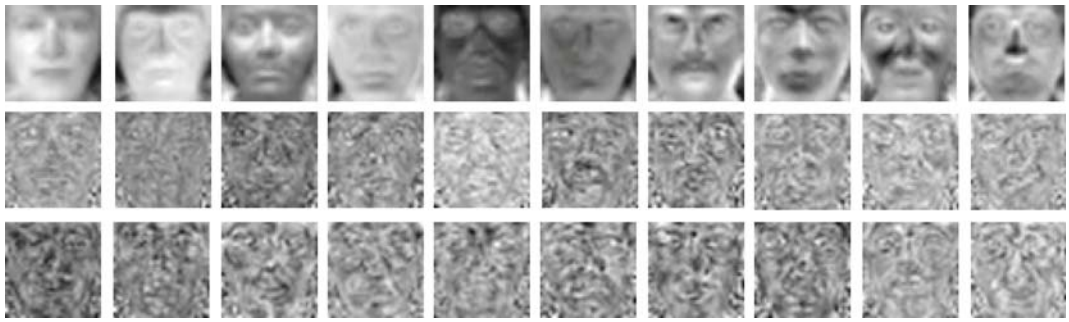


Fig. 2 The first 10 Eigenfaces (the first row), Fisherfaces (the second row), and Laplacianfaces (the third row) calculated from face images in YALE database

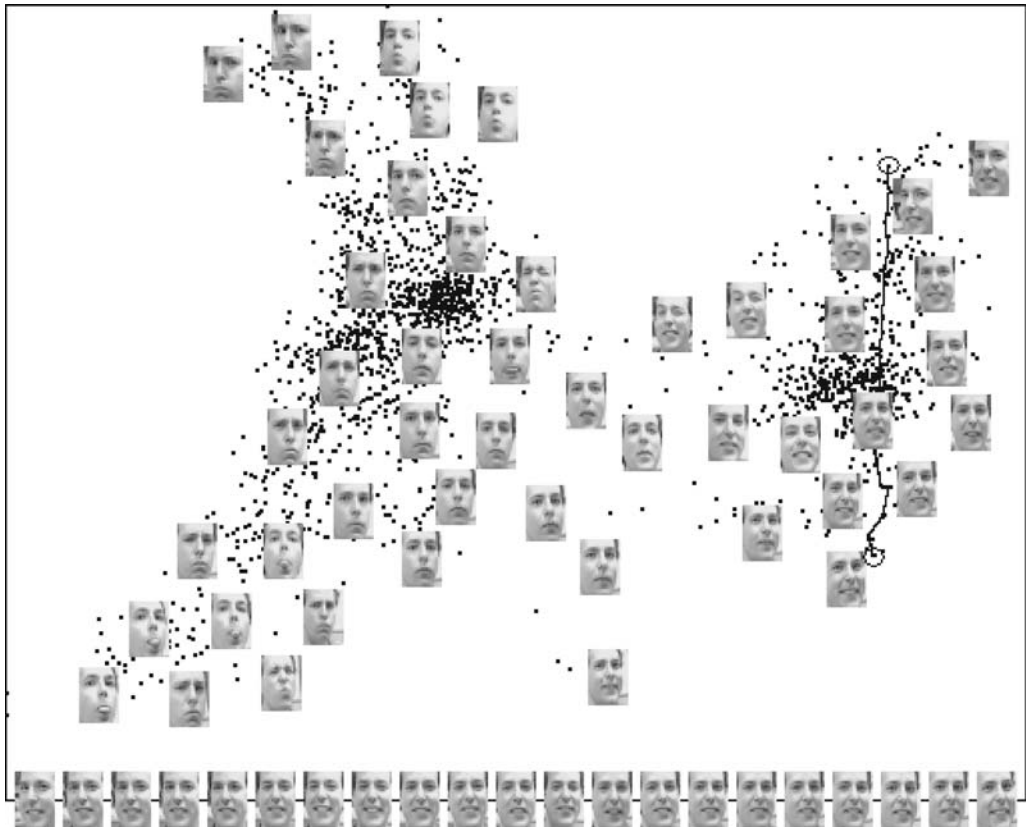


Fig. 3 2D linear embedding of face images by Laplacianfaces. As can be seen, the face images are divided into two parts: the faces with open mouth and the faces with closed mouth. Moreover, it can be clearly seen that the pose and expression of human faces change continuously and smoothly, from top to bottom and from left to right. The bottom images correspond to points along the right path (linked by solid line), illustrating one particular mode of variability in pose

mouth. This is because in trying to preserve local structure in the embedding, the Laplacianfaces implicitly emphasizes the natural clusters in the data. Specifically, it makes the neighboring points in the image face nearer in the face subspace, and faraway points in the image face farther in the face space. The bottom images correspond to points along the right path (linked by solid line), illustrating one particular mode of variability in pose. The 10 testing samples can be simply located in the reduced representation space by the Laplacianfaces (column vectors of the matrix $\boldsymbol{W}$). Figure 4 shows the result. As can be seen, these testing samples optimally find their coordinates which reflect their intrinsic properties, i.e., pose and expression. This observation tells us that the Laplacianfaces are capable of capturing the intrinsic face manifold structure to some extent.

2.2 Manifold regularized active learning

In this subsection, we introduce our work on active learning from a geometrical perspective.

2.2.1 Laplacian regularized D-optimal design for active learning

In many machine learning tasks, there is no shortage of unlabeled data, but labels are expensive to obtain. Thus, the problem of selecting the most informative data to label and learning from both labeled and unlabeled data emerge and attract considerable attention in recent years. The former is usually called *active learning*, while the latter is generally referred to as *semi-supervised learning*.

In statistics, the problem of selecting the most informative samples to label is usually called *experimental design*. The sample $\boldsymbol{x}$ is referred to as *experiment*, and its label y as *measurement*. The study of *optimal experimental design* (OED) [36] is concerned with the design of experiments that are expected to minimize variances of a parameterized model.

Specifically, consider a linear regression model

$$y = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + \epsilon, \tag{9}$$

where y is the *observation*, $\boldsymbol{x}$ is the *independent variable*, $\boldsymbol{w}$ is the *weight vector*, and ϵ is an unknown error with zero mean and variance σ^2. We define $f(x) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}$ to be the learner's output given input $\boldsymbol{x}$ and the weight vector $\boldsymbol{w}$. Suppose we have a set of m samples $X = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m)$, out of which l samples, $Z = (\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_l)$ are labeled, and y_i is the label of $\boldsymbol{z}_i$. Recently, Belkin et al. proposed a novel regression model called Laplacian regularized least squares (LapRLS) [14],

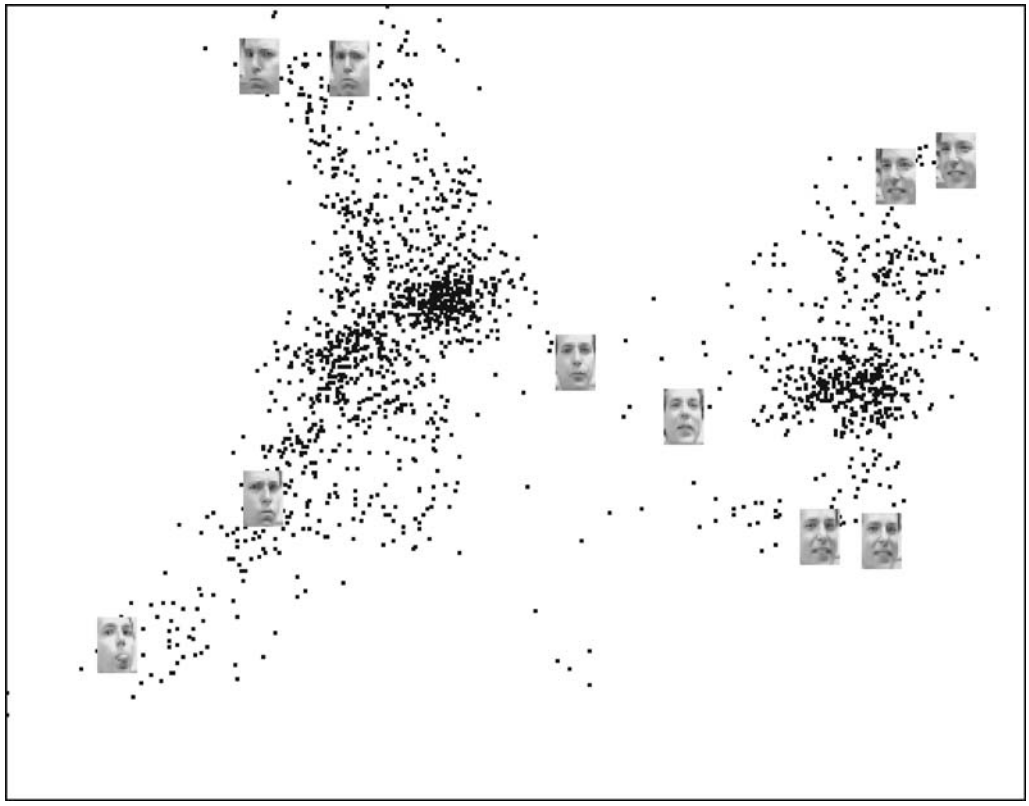


Fig. 4 Distribution of 10 testing samples in the reduced representation subspace. As can be seen, these testing samples optimally find their coordinates which reflect their intrinsic properties, i.e., pose and expression

which makes use of both labeled and unlabeled points. LapRLS minimizes the following objective function:

$$J(\boldsymbol{w}) = \sum_{i=1}^{l}(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i - y_i)^2 + \frac{\lambda_1}{2}\sum_{i,j=1}^{m}(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_j)^2 S_{ij} + \lambda_2\|\boldsymbol{w}\|^2, \tag{10}$$

whose solution is given by

$$\widehat{\boldsymbol{w}} = \left(ZZ^{\mathrm{T}} + \lambda_1 XLX^{\mathrm{T}} + \lambda_2 I\right)^{-1} Z\boldsymbol{y}. \tag{11}$$

We define $H = ZZ^{\mathrm{T}} + \lambda_1 XLX^{\mathrm{T}} + \lambda_2 I$ and $\Lambda = \lambda_1 XLX^{\mathrm{T}} + \lambda_2 I$. H is the hessian of $J(\boldsymbol{w})$. It is easy to show [15]:

$$E(\widehat{\boldsymbol{w}} - \boldsymbol{w}) = -H^{-1}\Lambda\boldsymbol{w}, \tag{12}$$

and

$$\mathrm{Cov}(\widehat{\boldsymbol{w}}) \approx \sigma^2 H^{-1}. \tag{13}$$

In order to minimize the bias of the estimator, as well as the size of the covariance matrix of the estimator, one has to minimize the size of the matrix H^{-1}. There are many different ways to measure the size of the covariance matrix, leading to different algorithms. The typical ones are A-optimality, which minimizes the trace of the covariance matrix; D-optimality, which minimizes the determinant of the covariance matrix; and E-optimality, which minimizes the largest eigenvalue of the covariance matrix [36]. In Refs. [15,16], we have introduced a novel active learning algorithm by using D-optimality, called Laplacian regularized D-optimal design (LapRDD). Noticing that $\det(H^{-1}) = 1/\det(H)$. The objective function is given as follows:

$$\max_{Z=(\boldsymbol{z}_1,\boldsymbol{z}_2,\ldots,\boldsymbol{z}_l)} \det(ZZ^{\mathrm{T}} + \lambda_1 XLX^{\mathrm{T}} + \lambda_2 I). \tag{14}$$

Please see Refs. [15,16] for the detailed optimization procedure.

2.2.2 A unified active and semi-supervised learning framework for image compression

We have applied our proposed active learning algorithm to image compression. Recently, Cheng et al. propose to treat image compression as a machine learning problem [37]. Instead of performing a frequency transformation, they store a grayscale version of the image and color labels of a few representative pixels. Using the stored information they apply LapRLS to learn a model that predicts the color for the rest of the pixels. There are two fundamental steps in machine-learning-based image compression: selecting the most representative pixels as encoding and colorization as decoding. The first step is essentially an active learning problem, while the second step is a semi-supervised learning problem.

We have applied our proposed LapRDD active learning algorithm, together with LapRLS, to image compression. Figure 5 shows an example. Figures 5(a) and 5(b) are the original image and the corresponding grayscale

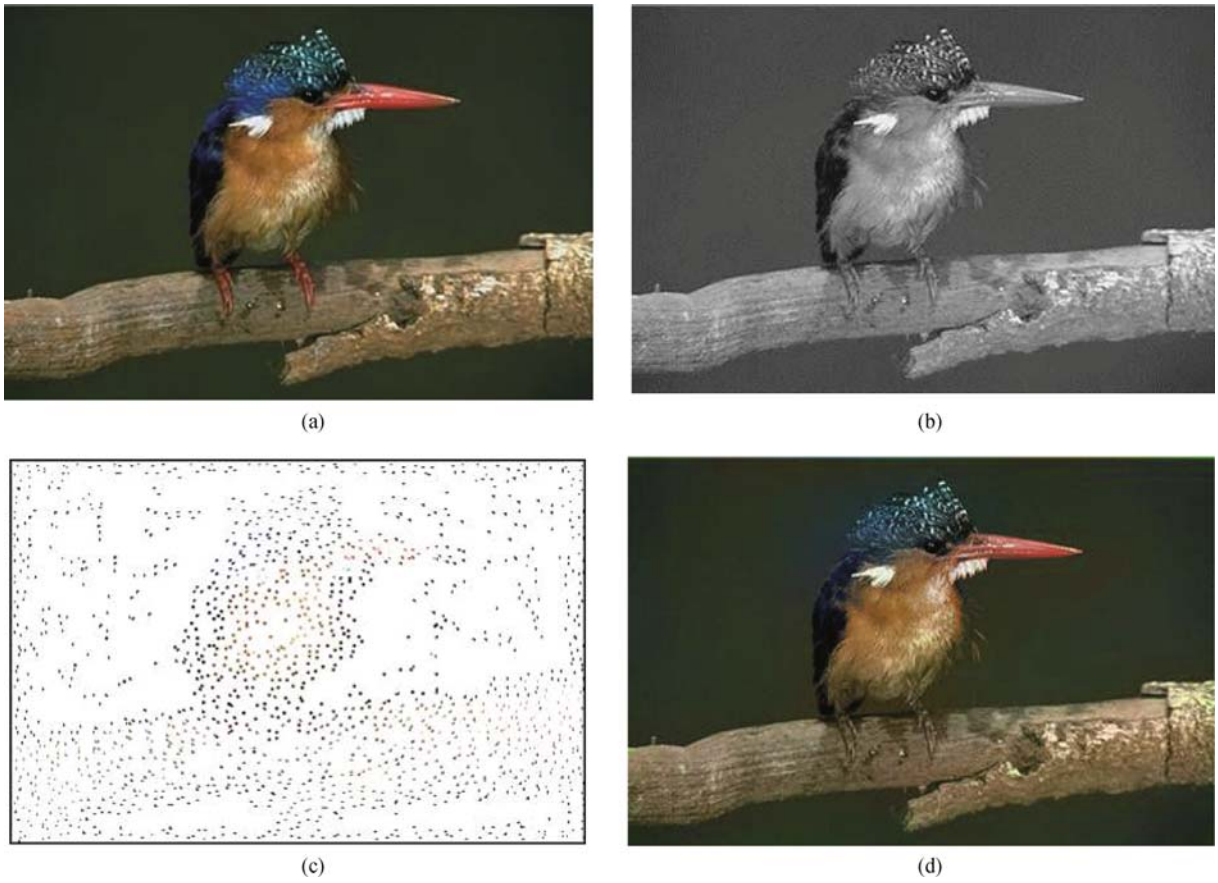


Fig. 5 Demonstration of image compression framework. (a) Original image; (b) grayscale image; (c) selected points; (d) recovered image. During encoding, the grayscale image (b) and selected points (c) are stored as compressed representation of the original image. During decoding, the image is reconstructed by using semi-supervised learning algorithm

image. Figure 5(c) shows the most informative pixels selected by our proposed LapRDD active learning algorithm. Figure 5(d) shows the uncompressed image by using semi-supervised learning algorithm.

3 Persistent tangent space learning

In this section, we introduce a novel algorithm for tangent space learning.

3.1 Problem

Estimating the tangent space locally is not new and difficult. Various local methods, for example local PCA, are developed. The main problem of local methods is that usually they are very sensitive to noise. Thus, how to choose suitable neighborhood size is crucial. When the neighborhood size is properly chosen, PCA can accurately estimate the tangent space even when there is some noise [38]. Consider that we perturb the entries of a matrix by adding independent Gaussian noise. If the variance of noise is not too big, then the optimal projections obtained by PCA are close to those of the original matrix without any noise. Thus, if we can choose a suitable neighborhood size, we can still find the tangent spaces by PCA.

Given a neighborhood size at a point, a tangent space can be computed. If the neighborhood size varies, we get a sequence of tangent spaces indexed by neighborhood size. Thus, here comes the problem on how to choose the tangent space. The intuitive idea here is that we can measure the change of the tangent space sequence. We can choose the tangent space in a stable period of neighborhood size that the tangent spaces vary little. Figure 6 shows an illustrative example. We apply local PCA at the 28th point denoted by green circle. Figure 6(b) shows the result of local PCA with different numbers of nearest neighbors. As we can see, when the neighborhood size is too small, estimated tangent space is biased by noise. When the neighborhood size is too large, estimated tangent space is biased by the global structure of the manifold. Figure 6(c) shows the *change* of tangent space sequences of four different points denoted by colored circle. When the neighborhood size is small, the change of tangent space is very high. Then, it archives relatively small change. When the neighborhood size becomes larger, there is a jump here. Although the change of tangent space sequence is very low, it is highly biased

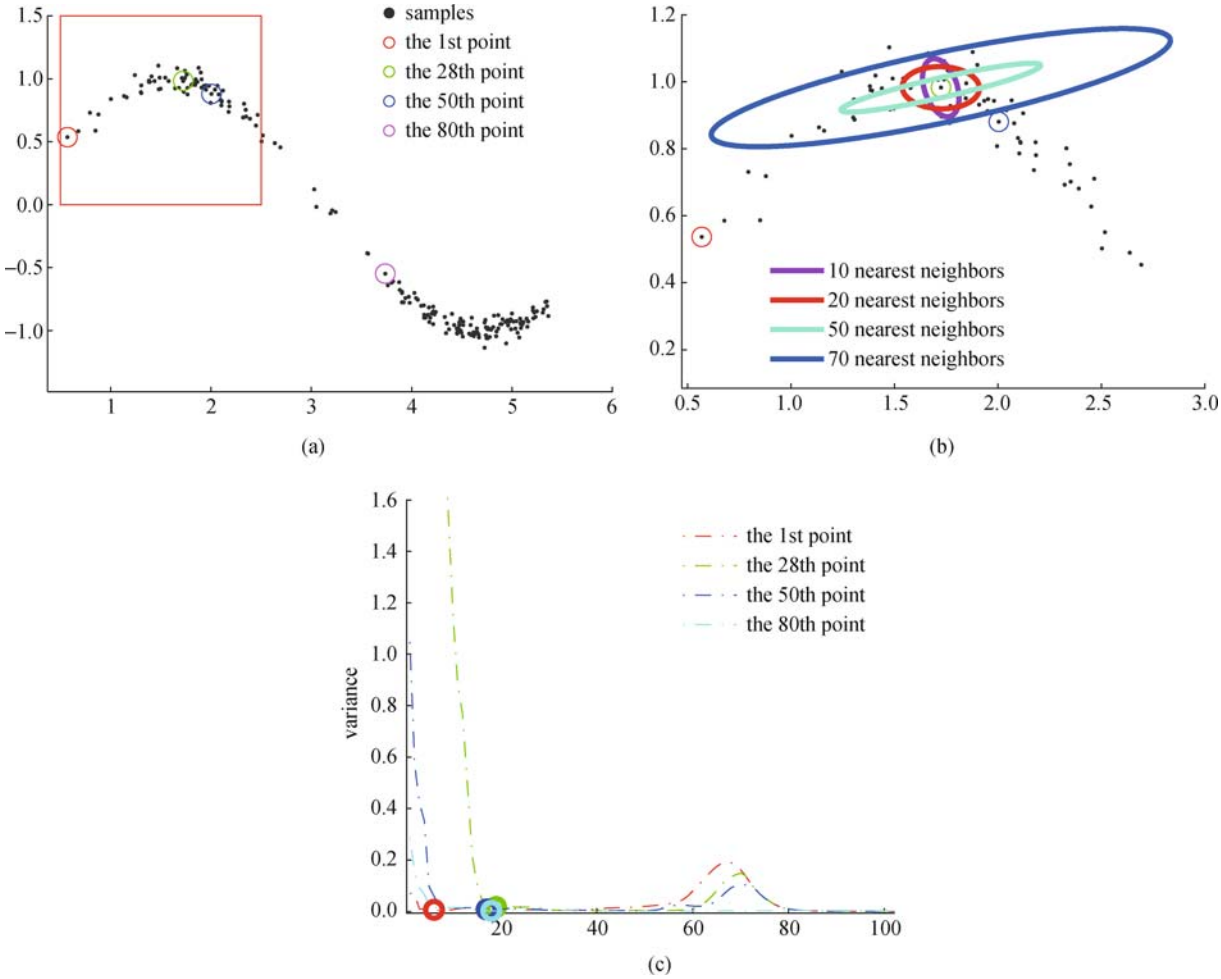


Fig. 6 Illustration of our approach. (a) Sin curve. 200 points are sampled unevenly from an one-dimensional sin curve ($d = 1$) embedded in a 2D space ($m = 2$) with Gaussian noise ($\sigma = 0.05$). In this example, we apply local PCA at the 28th point; (b) results of local PCA with different numbers of nearest neighbors. Then, we compute variance curve at each data point; (c) variance curves of four different points denoted by colored circle. Although they have different curvature and density, they have similar variance curve

by the global structure. Thus, intuitively, we should choose the first low change period. We provide formal description below.

Given $x_i \in \mathbb{R}^m$, $i = 1, 2, \ldots, n$, which are sampled from a low-dimensional manifold $\mathcal{M} \subset \mathbb{R}^m$. We aim to find the tangent spaces at each point. If we choose a suitable neighborhood size, PCA can be applied to estimate the proper tangent space. In this paper, we will consider each tangent space as a point on a Grassmann manifold; thus, we can deal with tangent space directly on Grassmann manifold. For each x_i, denote $H_j(x_i)$ be the result of PCA with $j-1$ nearest points. Thus, our problem can be formulated as follows: given a sequence of tangent spaces $H_{d+1}(x_i), H_{d+2}(x_i), \ldots, H_n(x_i)$, find the first stable period in this tangent spaces sequence.

3.2 Grassmann manifold and tangent space

We begin with an introduction of some concepts from differential geometry and statistics on Riemannian geometry which we will work with.

Definition 1 The Grassmann manifold $\mathcal{G}(d, m)$ is the set of d-dimensional linear subspaces of the $\mathbb{R}^m$.

The Grassmann manifold [39] $\mathcal{G}(d, m)$ is a $d(m-d)$-dimensional compact Riemannian manifold. At each point x in $\mathcal{M}$, the tangent space to $\mathcal{M}$ can be considered as a subspace of $\mathbb{R}^m$, which is just a point in $\mathcal{G}(d, m)$. This defines a map from $\mathcal{M}$ to Grassmann manifold $\mathcal{G}(d, m)$. For tangent space learning, we are trying to find such a map, $H : \mathcal{M} \to \mathcal{G}$. The point in Grassmann manifold can be represented by an $m \times d$ orthonormal matrix Y such that $Y'Y = I_d$, where I_d is the $d \times d$ identity matrix. However, such matrix representation is not unique. If two matrices Y_1 and Y_2 span the same subspace by their column vectors, i.e., $\text{span}(Y_1) = \text{span}(Y_2)$, then they are considered as the same point on Grassmann manifold. In other words, $\text{span}(Y_1) = \text{span}(Y_2)$ if and only if $Y_1 R_1 = Y_2 R_2$ for some $R_1, R_2 \in \mathcal{O}(d)$, where $\mathcal{O}(d)$ denotes the set of $d \times d$ orthonormal matrices. Therefore, we will use Y to denote its equivalence class $\text{span}(Y)$, and use $Y_1 = Y_2$ to

denote $\text{span}(Y_1) = \text{span}(Y_2)$, for the sake of simplicity.

Since we consider tangent space as a point in Grassmann manifold, we can use the Riemannian metric in Grassmann manifold to measure the distance between tangent spaces. A Riemannian metric is a function that satisfies the following axioms:

Definition 2 A real-valued function $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a metric if

1) $d(x_1, x_2) \geqslant 0$,

2) $d(x_1, x_2) = 0$ if and only if $x_1 = x_2$,

3) $d(x_1, x_2) = d(x_2, x_1)$,

4) $d(x_1, x_2) + d(x_2, x_3) \leqslant d(x_1, x_3)$ for all $x_1, x_2, x_3 \in \mathcal{X}$.

A distance (or a metric) between tangent spaces $d(Y_1, Y_2)$ has to be invariant under different representations $d(Y_1, Y_2) = d(Y_1R_1, Y_2R_2)$, for any $R_1, R_2 \in \mathcal{O}(m)$. Formally, the Riemannian distance between two tangent spaces is the length of the shortest geodesic connecting the two points on the Grassmann manifold.

Definition 3 The projection distance is defined as

$$d_{\text{proj}}(Y_1, Y_2) = 2^{-1}\|Y_1Y_1' - Y_2Y_2'\|_{\text{F}}^2.$$

In this paper, we use the projection distance [39] to measure the distance between two tangent spaces. The reason we choose projection distance is due to the property that it is invariant under different representations, and it is easy to compute. More importantly, the statistics on Grassmann manifold becomes tractable by using projection distance.

3.3 Statistics on Grassmann manifold

Next, we introduce the statistics on Riemannian manifold. A natural choice for computing the "center" in a metric space $\mathcal{M}$ is the Fréchet mean. Let Q be a probability measure on $\mathcal{M}$, let d be a metric on $\mathcal{M}$, the Fréchet mean minimizes $F(p) = \int d^2(p, x)Q(dx)$, if there is a unique minimizer. In general, the set of all minimizers is called the *Fréchet mean set*.

When $\mathcal{M}$ is a complete d-dimensional Riemannian manifold, we will refer to the Fréchet mean (set) as the intrinsic mean (set). The intrinsic mean exists if there is a unique minimizer.

Definition 4 (Intrinsic mean set [40]). Let Q be a probability measure on $\mathcal{M}$. The intrinsic mean set of Q is the Fréchet mean set of Q w.r.t the distance d_g. If there is a unique minimizer, this is called the intrinsic mean of Q, denoted by $\mu_i(Q)$, or μ_i. If X is an $\mathcal{M}$-valued random variable having distribution Q, then the above is also referred to as the intrinsic mean (set) of X.

Definition 5 (Intrinsic total variance [40]). If Q is a probability measure on $\mathcal{M}$, the intrinsic total variance of Q, tV_i, is the value of the Fréchet function F at a point q in the intrinsic mean set of Q, provided $F(p) < \infty$ for some $p \in \mathcal{M}$. In particular, if the intrinsic mean exists, then $tV_i = F(\mu_i)$.

As we only have finite samples, here, we introduce *intrinsic sample mean* and *intrinsic total sample variance*.

Definition 6 (Intrinsic sample mean [40]). Let $X_1, X_2, \ldots, X_n$ be independent random variables with a common distribution Q and consider their empirical distribution $\hat{Q}_n = \sum_{k=1}^n \delta_{X_k}/n$. The intrinsic sample mean (set) is the intrinsic mean (set) of $\hat{Q}_n$, i.e., the (set of) minimizer(s) of $p \to \sum_{j=1}^n d_g^2(X_j, p)/n$.

Definition 7 (Intrinsic sample total variance [40]). Let $X_1, X_2, \ldots, X_n$ be independent random variables with a common distribution Q. The intrinsic total sample variance $tV_i(\hat{Q}_n)$ is the intrinsic total variance of the empirical $\hat{Q}_n$.

In the case when the manifold is Grassmann manifold with projection distance on it, we have the following theorem.

Theorem 2 Let $\mathcal{M}$ be a Grassmann manifold $\mathcal{G}(d, m)$, let d_{proj} be the projection distance on it; then, the intrinsic sample mean (set) and intrinsic sample total variance exist.

Proof Consider the following minimization problem:

$$\min F(H) = \sum_{i=1}^n \frac{1}{n} d_{\text{proj}}(H, H_i)^2.$$

By the definition of projection distance, we have

$$F(H) = \frac{1}{n}\sum_{i=1}^n \|HH' - H_iH_i'\|_{\text{F}}^2.$$

By the definition of Frobenius norm, the above optimization problem is equivalent to the following:

$$\left\| HH' - \frac{1}{n}\sum_{i=1}^n H_iH_i' \right\|_{\text{F}}^2.$$

Decompose $\sum_i H_iH_i'/n$ by singular value decomposition, we have $\sum_i H_iH_i'/n = U\Sigma U'$, where U is $m \times m$ unitary matrix, and Σ is a diagonal matrix $\Sigma = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_m)$. Without loss of generality, let $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_m$. Thus, the minimization problem becomes

$$\min \|HH' - U\Sigma U'\|_{\text{F}}^2.$$

According to Eckart-Young theorem [41], we have $HH' = USU'$, where $S = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_d, 0, \ldots, 0)$. If all singular values are different, we have $H = U\sqrt{S}$, and this expression is unique. If there are multiple singular values in $\{\lambda_1, \lambda_2, \ldots, \lambda_d\}$, the expression of H is not unique but dependent on the choice of basis. However, they still represent the same tangent space. If there are multiple singular values $\{\lambda_{d+1}, \lambda_{d+2}, \ldots, \lambda_{d+s}\}$ that are equal to λ_d, then H is not unique.

In conclusion, the intrinsic sample mean (set) exists and intrinsic sample total variance equals $F(H) = F(U\sqrt{S})$.

3.4 Algorithm

For each x_i, we need to find an appropriate neighborhood for estimating the tangent space. With different size of local neighborhood, we can estimate different tangent spaces. The best neighborhood size k is estimated by using the *intrinsic sample total variance* to measure the persistence. We call it PTSL and give the detailed steps below.

For each x_i, we compute tangent spaces via local PCA. The feasible neighborhood size can be $d+1, d+2, \ldots, n$ because with less points estimating a d-dimensional tangent space is an ill-posed problem. For each neighborhood size, we can compute the corresponding tangent space, denoted by $H_{d+1}(x_i), H_{d+2}(x_i), \ldots, H_n(x_i)$. The variation of the tangent space sequences follows some patterns: At first, the tangent spaces are rather unstable because of noises. When the neighborhood size increase to a sufficient number, the estimation of the tangent space would approximate the *true* value. This approximation will keep stable as the neighborhood size continues to increase until the global structure is involved. After that, the estimated tangent space sequence might vary drastically or keep relatively stable depending on the overall shape of the underlying manifold. However, in any case, a final stable period will be reached when the estimated tangent space is dominated by the global geometry of the manifold, which is actually the result of global PCA.

Since our goal is to find an approximation of the *true* tangent space, we need to find the first stable neighborhood size. Specifically, for a neighborhood size k, we compute the *intrinsic sample total variance* of $H_k, H_{k+1}, \ldots, H_{k+p-1}$ and use it (denoted by V_k) to measure the stableness of k. Here, p is a fixed window size. The smaller V_k is, the more stable the neighborhood as well as the corresponding tangent space is.

Then, the problem of finding the first stable neighborhood size k is transformed to finding the first small V_k. Simply finding the minimum V_k does not work because the remaining stable period corresponding to the global geometry of manifold might give smaller V. Intuitively, we need to find the first local minimum if we consider (k, V_k) as a curve. The procedure of searching the first local minimum is very simple: we start at the first point, keep increasing, and break until V stops to decrease. Then, the corresponding neighborhood size k is what we desire. However, this simple procedure is sensitive to random noises, which may create false local minimums. Therefore, we use two threshold parameters to filter out noises. The final procedure is described below:

- Start with $j = d+1$ and set $V_{\min} = V_j$ and $j_{\min} = j$.
- Loop
  - Set $j \leftarrow j + 1$;
  - If $V_j < \gamma_1 V_{\min}$, then update $j_{\min} \leftarrow j$ and $V_{\min} \leftarrow V_j$;
  - If $V_j > \gamma_2 V_{\min}$, break the loop.
- $k = j_{\min}$ is the desired neighborhood size.

Figure 6 gives an illustrative example for this process. Finally, the desired tangent space is given by the intrinsic sample mean of $H_k, H_{k+1}, \ldots, H_{k+p-1}$ (or we can also simply set it to be H_k).

In the ideal case, both of the parameters γ_1 and γ_2 equal to 1. However, when there are noise and outliers present, smaller γ_1 and larger γ_2 should be better. In all our experiments, we set γ_1 and γ_2 to 0.1 and 2, respectively.

3.5 Generalization

Given a set of data points $x_i \in \mathbb{R}^m, i = 1, 2, \ldots, n$ and their tangent spaces $H_i \in \mathbb{R}^{m\times d}, i = 1, 2, \ldots, n$. Our goal is to find the best estimated tangent space H on a novel data point x. This problem can be formalized as a supervised learning problem. Given training set $(x_i, H_i), i = 1, 2, \ldots, d$, we are trying to find a map $f : \mathbb{R}^m \to \mathcal{G}(d, m)$. This problem is different from traditional learning problem since the range of this mapping is Grassmann manifold rather than Euclidean space. This may incur problem since we even can not use linear combination of tangent spaces. Intuitively, we can let $f(x) = \sum_i H_i(x_i)w(x, x_i)$, $\sum_i w(x, x_i) = 1$. However, this function is meaningless since linear combination is not defined on Grassmann manifold.

Consequently, we have to consider linear combination in manifold setting. In this way, the above linear combination can be viewed as a minimization problem:

$$\min F(H) = \sum_{i=1}^{n} d_{\text{proj}}(H, H_i)^2 w_i(x, x_i),$$
$$\sum_i w_i(x, x_i) = 1. \tag{15}$$

This is exactly the intrinsic mean on Grassmann manifold, where $w_i(x, x_i), i = 1, 2, \ldots, n$, are the corresponding probabilities of H_i. Thus, the tangent space at new point can be viewed as mean of tangent spaces closest to it. In this paper, we use heat kernel as weights $w_i(x, x_i) = \mathrm{e}^{-\frac{\|x-x_i\|^2}{t}}/c$, where $c = \sum_i \mathrm{e}^{-\frac{\|x-x_i\|^2}{t}}$ is the normalization factor. According to Theorem 2, the intrinsic sample mean (set) exists, and we have

$$H = U\text{diag}(\sqrt{\lambda_1}, \sqrt{\lambda_2}, \ldots, \sqrt{\lambda_d}, 0, \ldots, 0),$$

where $\sum_i^n w_i M_i = U\Sigma U'$.

Note that, some nonlinear interpolation methods can also be applied here. Likewise, we have to find the nonlinear interpolation in manifold setting that should be formulated as a nonlinear minimization problem on the manifold. This can be quite challenging since there is no standard way to solve a nonlinear optimization problem on Grassmann manifold.

4 Experimental result

In this section, we give several experimental results on both synthetic data and real data.

4.1 Synthetic examples

In this section, we compare the performance of different algorithms by generating synthetic manifolds and compare the calculated tangent space with the ground truth. Two examples are presented in Fig. 7. In Figs. 7(a) and 7(b), 200 points from a one-dimensional sin curve ($d = 1, m = 2$) are sampled as test data. The data points are sampled unevenly and added with Gaussian noise ($\sigma = 0.05$). The results of LSML and PTSL are shown in Figs. 7(a) and 7(b), respectively, with blue points representing samples and green arrows the learned tangent spaces. From this example, we can see that our approach performs much better than LSML.

Figures 7(c) and 7(d) show an example on a half-sphere ($d = 2, m = 3$). Figure 7(c) shows the 100 samples generated from the 2D manifold in a similar manner ($\sigma = 0.05$ Gaussian noise and uneven sample). We compare the calculated tangent space to the ground truth with the *projection distance* and compute the average error. Three algorithms are compared in Fig. 7(d). Note that, the value of k in LSML is fixed to 20. As can be

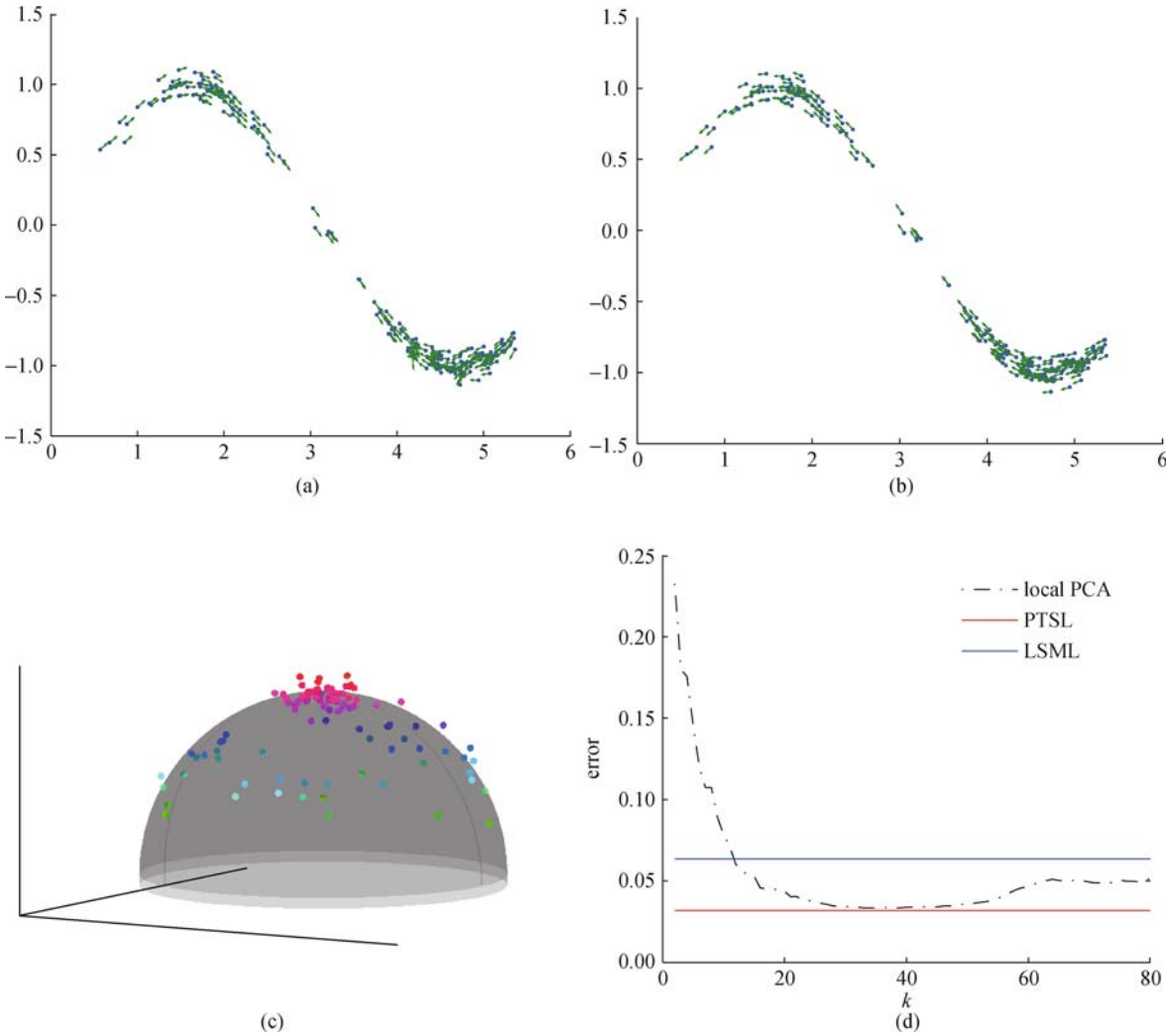


Fig. 7 Comparison between LSML and PTSL on tangent space learning. (a) Result of LSML on sin data; (b) result of PTSL on sin data; (c) 100 sampled points on a half-sphere; (d) error (measured by projection distance to the ground truth) comparison for half-sphere data of the three algorithms. Synthetic manifold samples are used to evaluate the algorithms. Data points are sampled unevenly and added with Gaussian noise ($\sigma = 0.05$). In (a) and (b), 200 points from a one-dimensional sin curve ($d = 1, m = 2$) are sampled as test data. The blue points represent samples and green arrows the learned tangent spaces. In (c) and (d), we show the results of a half-sphere ($d = 2, m = 3$). Note that, the value of k in LSML is fixed to 20

seen, local PCA achieves its best when the neighborhood size is around 30∼40. With either smaller or larger neighborhood size, its performance gets worse. While LSML performs not very well in this case, PTSL has a lower error rate than even the best of local PCA, which verified the necessity of choosing different neighborhood sizes at different points.

4.2 Generalization result

This experiment examines the performance of generalization of PTSL. We sample training data points from a circle without any noise. Once the tangent spaces on the training data points are obtained, we can use the method discussed above to estimate the tangent spaces at novel points. In this example, we consider those points inside the circle. Specifically, we apply the LSML and PTSL algorithms to learn tangent spaces on 100 training data points sampled from unit circle. We then randomly generate 20 data points inside this circle for test. The tangent vectors obtained by LSML and PTSL are shown in Figs. 8(a) and 8(b), respectively. The green vectors denote the tangent vectors of the training points on the circle, and the red vectors denote the tangent vectors of the test points inside the circle. Figure 8(a) indicates that LSML fails to estimate the tangent vectors on most of novel data points. The directions of these tangent vectors are twisted greatly and do not form concentric circles, whereas PTSL gives much better results, as shown in Fig. 8(b). Figures 8(c) and 8(d) shows the normal vectors obtained by LSML and PTSL on a hemisphere, respectively. In this example, 200 random points are sampled from the hemisphere. The different colors correspond to different distances from the center. As can be seen, PTSL performs much better than LSML.

4.3 Tangent spaces for clustering

The tangent space information can be used to help data clustering. To be specific, by requiring data points within the same cluster having similar tangent spaces, we can better explore the shape and local linearity of the clusters. We consider this problem in the product space $\mathbb{R}^n \times \mathcal{G}(d, m)$. Thus, each point x endowed with a tangent space H_x in this new space is represented by (x, H_x). Since the Euclidean space $\mathbb{R}^n$ and the

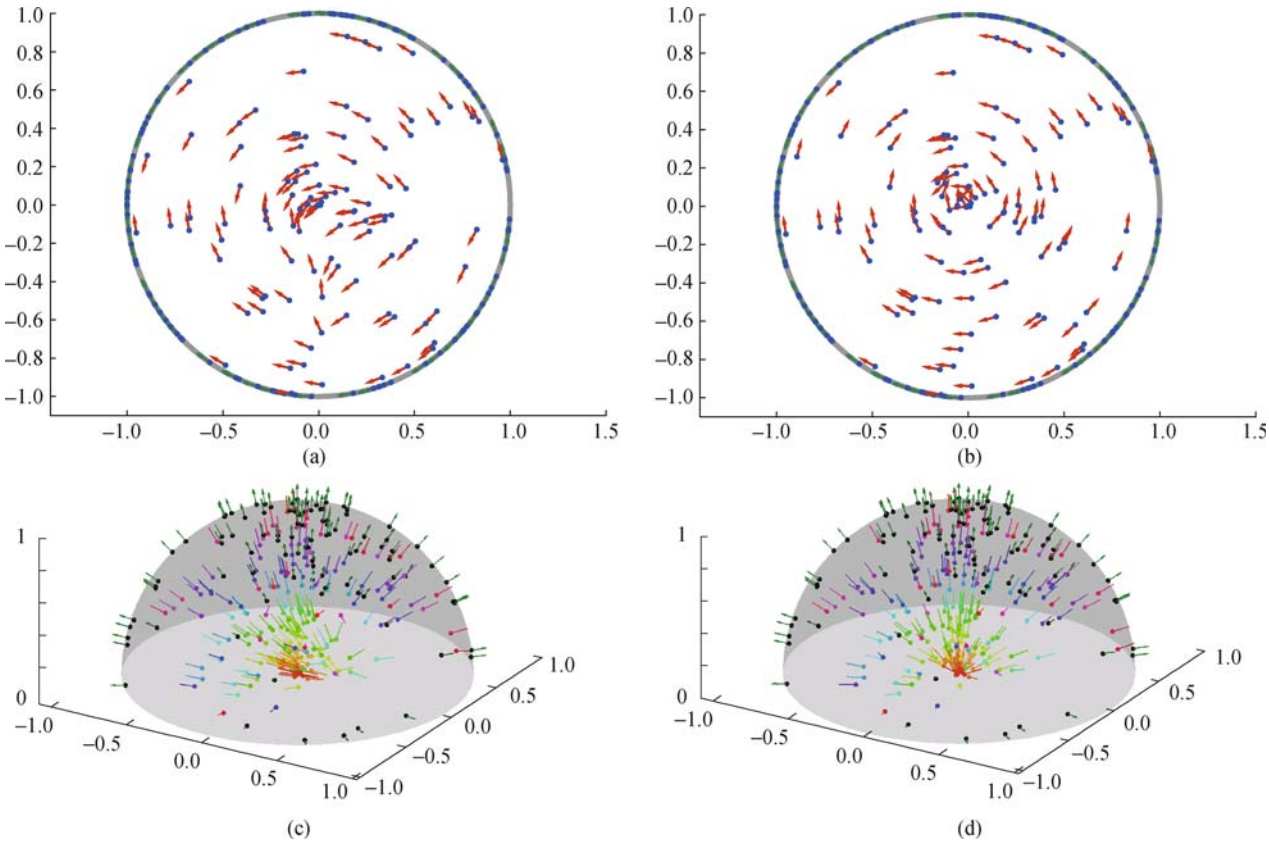


Fig. 8 Generalization performance comparison on unit circle and hemisphere between LSML and PTSL. (a) Tangent vectors obtained by LSML algorithm on unit circle (the green vectors denote the tangent vectors of the training points on the circle, and the red vectors denote the tangent vectors of the test points inside the circle); (b) tangent vectors obtained by PTSL algorithm on unit circle; (c) normal vectors obtained by LSML algorithm on hemisphere (the different colors correspond to different distances from the center); (d) normal vectors obtained by PTSL algorithm on hemisphere. The training set consists of 100 points sampled from 1D circle and 200 points from the hemisphere, which are embedded in a 2D space and 3D space, respectively. The test points are randomly generated inside unit circle and the hemisphere

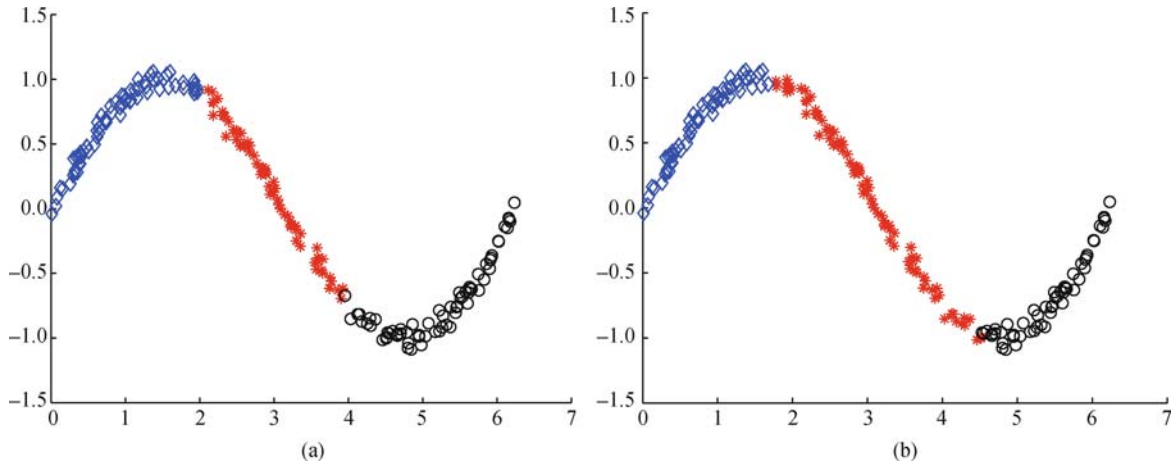


Fig. 9 Clustering results. (a) Result of standard Kmeans (Different colors represent different clusters); (b) result of PTS Kmeans by using the tangent space information ($\lambda = 25$). We sampled 200 points unevenly from a sin curve with Gaussian noise ($\sigma = 0.05$) and then cluster them into three segments

Grassmann manifold $\mathcal{G}(d, m)$ are independent, we define the new distance by combining the standard Euclidean distance with projection distance as follows:

$$d_{\text{new}}(x, y) = \sqrt{\|x - y\|_2^2 + \lambda d_{\text{proj}}(H_x, H_y)^2}. \quad (16)$$

The parameter λ measures the importance of tangent space information. Thus, the traditional K-means algorithm can be performed by using the new distance metric. We call it PTS Kmeans. We sampled 200 points unevenly from a sin curve with Gaussian noise ($\sigma = 0.05$) and then cluster them into three segments by standard Kmeans and PTS Kmeans, respectively. To avoid local minimum of Kmeans, both algorithms are performed 20 times, and the best results (measured by the Kmeans objective function) are chosen. The results are shown in Fig. 9. Comparing with the standard Kmeans, our algorithm successfully recovers the three nearly linear segments of the sin curve, and the obtained clusters are more natural with respect to the manifold structure.

5 Conclusion

In this paper, we consider two fundamental problems in tangent space learning. One is how to estimate tangent space on finite data samples. Different from previous work on tangent space learning, we consider the space of tangent spaces as a Grassmann manifold, and we use the statistics on Grassmann manifold to deal with tangent spaces. We have also considered the generalization problem of tangent space learning. It turns out to be the problem of computing the mean on Grassmann manifold. The results of our persistent tangent space learning are impressive. As our experiments demonstrate, the learned tangent space can be used to calculate the distance between data points, and thus, better clustering performance can be obtained by using the new distance metric.

The algorithm presented in this paper is essentially a local one in that the tangent space at each local neighborhood is computed independently. The advantage is that it is very computationally efficient since it avoids solving a global optimization problem as most of previous methods do. However, in some situations, it might be necessary to require the learned tangent spaces varying as smoothly as possible along the geodesics of the manifold. It remains unclear how to explicitly ensure the smoothness without the increase of computational cost. This problem is left for our future work.

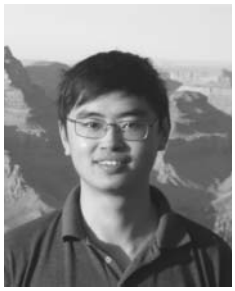
Acknowledgements This work was supported by the National Natural Science Foundation of China (Grant No. 60875044).

References

1. Pearson K. On lines and planes of closest fit to systems of points in space. Philosophical Magazine, 1901, 2(6): 559–572
2. Schölkopf B, Smola A, Müller K R. Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation, 1998, 10(5): 1299–1319
3. Ham J, Lee D D, Mika S, Schölkopf B. A kernel view of the dimensionality reduction of manifolds. In: Proceedings of the 21st International Conference on Machine Learning. 2004, 369–276
4. Tenenbaum J, de Silva V, Langford J. A global geometric framework for nonlinear dimensionality reduction. Science, 2000, 290(5500): 2319–2323
5. Roweis S, Saul L. Nonlinear dimensionality reduction by locally linear embedding. Science, 2000, 290(5500): 2323–2326
6. Belkin M, Niyogi P. Laplacian eigenmaps and spectral techniques for embedding and clustering. In: Proceedings of Advances in Neural Information Processing Systems. 2001, 14: 585–591
7. de Silva V, Tenenbaum J B. Global versus local methods in nonlinear dimensionality reduction. In: Becker S, Thrun

S, Obermayer K, eds. Proceedings of Advances in Neural Information Processing Systems. 2002, 15: 721–728
8. Zhang Z, Wang J. MLLE: Modified locally linear embedding using multiple weights. In: Schölkopf B, Platt J, Hoffman T, eds. Proceedings of Advances in Neural Information Processing Systems. 2007, 19: 1593–1600
9. Weinberger K Q, Sha F, Saul L K. Learning a kernel matrix for nonlinear dimensionality reduction. In: Proceedings of the 21th International Conference on Machine Learning. 2004, 106–113
10. Coifman R R, Lafon S, Lee A B, Maggioni M, Warner F, Zucker S. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. In: Proceedings of the National Academy of Sciences. 2005, 7426–7431
11. Chung F R K. Spectral Graph Theory. Regional Conference Series in Mathematics Vol 92. Providence: American Mathematical Society, 1997
12. Bengio Y, Paiement J, Vincent P, Delalleau O, Roux N L, Ouimet M. Out-of-sample extensions for LLE, Isomap, MDS, eigenmaps, and spectral clustering. In: Thrun S, Saul L, Schölkopf B, eds. Proceedings of Advances in Neural Information Processing Systems. 2004, 16: 177–184
13. Belkin M, Niyogi P. Convergence of Laplacian eigenmaps. In: Schöllkopf B, Platt J, Hoffman T, eds. Proceedings of Advances in Neural Information Processing Systems. Cambridge: MIT Press, 2007, 19: 129–136
14. Belkin M, Niyogi P, Sindhwani V. Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. Journal of Machine Learning Research, 2006, 7: 2399–2434
15. He X, Ji M, Bao H. A unified active and semi-supervised learning framework for image compression. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition. 2009, 65–72
16. He X. Laplacian regularized d-optimal design for active learning and its application to image retrieval. IEEE Transactions on Image Processing, 2010, 19(1): 254–263
17. He X, Niyogi P. Locality preserving projections. In: Proceedings of Advances in Neural Information Processing Systems. 2003, 153–160
18. He X, Yan S, Hu Y, Niyogi P, Zhang H J. Face recognition using Laplacianfaces. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005, 27(3): 328–340
19. He X, Cai D, Niyogi P. Tensor subspace analysis. In: Proceedings of Advances in Neural Information Processing Systems. 2005, 18: 499–506
20. Donoho D L, Grimes C. Hessian eigenmaps: locally linear embedding techniques for high-dimensional data. Proceedings of the National Academy of Sciences, 2003, 100(10): 5591–5596
21. Steinke F, Hein M. Non-parametric regression between manifolds. In: Koller D, Schuurmans D, Bengio Y, Bottou L, eds. Proceedings of Advances in Neural Information Processing Systems. 2008, 21: 1561–1568
22. Kim K I, Steinke F, Hein M. Semi-supervised regression using hessian energy with an application to semi-supervised dimensionality reduction. In: Bengio Y, Schuurmans D, Lafferty J, Williams C K I, Culotta A, eds. Advances in Neural Information Processing Systems. 2009, 22: 979–987
23. Zhang Z, Zha H. Principal manifolds and nonlinear dimension reduction via local tangent space alignment. SIAM Journal of Scientific Computing, 2004, 26(1): 313–338
24. Brand M. Charting a manifold. In: Proceedings of Advances in Neural Information Processing Systems. 2003, 15: 961–968
25. Lin T, Zha H. Riemannian manifold learning. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2008, 30(5): 796–809
26. Dollár P, Rabaud V, Belongie S. Non-isometric manifold learning: analysis and an algorithm. In: Proceedings of the 24th International Conference on Machine learning. 2007, 227: 241–248
27. Dollár P, Belongie S, Rabaud V. Learning to traverse image manifolds. In: Schölkopf B, Platt J, Hoffman T, eds. Proceedings of Advances in Neural Information Processing Systems. 2007, 19: 361–368
28. Zomorodian A, Carlsson G. Computing persistent homology. Discrete and Computational Geometry, 2005, 33(2): 249–274
29. Niyogi P, Smale S, Weinberger S. Finding the homology of submanifolds with high confidence from random samples. Discrete and Computational Geometry, 2008, 39(1): 419–441
30. Eells J, Lemaire L. Selected Topics in Harmonic Maps. CBMS Regional Conference Series in Mathematics, Vol 50. Providence: American Mathematical Society, 1983
31. Bengio Y, Monperrus M. Non-local manifold tangent learning. In: Proceedings of Advances in Neural Information Processing Systems. 2005, 17: 129–136
32. Cai D, He X, Han J. Document clustering using locality preserving indexing. IEEE Transactions on Knowledge and Data Engineering, 2005, 17(12): 1624–1637
33. Lin B, He X, Zhou Y, Liu L, Lu K. Approximately harmonic projection: theoretical analysis and an algorithm. Pattern Recognition, 2010, 43(10): 3307–3313
34. Turk M, Pentland A. Eigenfaces for recognition. Journal of Cognitive Neuroscience, 1991, 3(1): 71–86
35. Belhumeur P N, Hepanha J P, Kriegman D J. Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1997, 19(7): 711–720
36. Atkinson A C, Donev A N. Optimum Experimental Designs. Oxford: Oxford University Press, 2007
37. Cheng L, Vishwanathan S V N. Learning to compress images and videos. In: Proceedings of the 24th International Conference on Machine Learning. 2007, 161–168
38. Achlioptas D. Random matrices in data analysis. In: Proceedings of the 15th European Conference on Machine Learning. 2004, 1–8
39. Ham J, Lee D D. Grassmann discriminant analysis: a unifying view on subspacebased learning. In: Proceedings of the 25th International Conference on Machine Learning. 2008, 376–383
40. Bhattacharya R, Patrangenaru V. Nonparametic estimation of location and dispersion on riemannian manifolds. Journal of Statistical Planning and Inference, 2002, 108: 23–36

41. Eckart C, Young G. The approximation of one matrix by another of lower rank. Psychometrika, 1936, 1(3): 211–218



Xiaofei He received the B.S. degree in Computer Science from Zhejiang University of China in 2000, and the Ph.D. degree in Computer Science from the University of Chicago in 2005. He is a Professor in the State Key Lab of CAD&CG at Zhejiang University, China. Prior to joining Zhejiang University in 2007, he was a Research Scientist at Yahoo! Research Labs, Burbank, CA. His research interests include machine learning, information retrieval, and computer vision. He is a senior member of IEEE.

Binbin Lin received the B.S. degree in Mathematics from Zhejiang University of China in 2007. He is currently working toward the Ph.D. degree in the Department of Computer Science, Zhejiang University. His research interests are manifold learning, statistical learning theory and kernel methods.