Neural Processing Letters manuscript No.
(will be inserted by the editor)

Local Tangent Space Discriminant Analysis

Yang Zhou · Shiliang Sun

Received: date / Accepted: date

Abstract We propose a novel supervised dimensionality reduction method named *local Tangent Space Discriminant analysis* (TSD) which is capable of utilizing the geometrical information from tangent spaces. The proposed method aims to seek an embedding space where the local manifold structure of the data belonging to the same class is preserved as much as possible, and the marginal data points with different class labels are better separated. Moreover, TSD has an analytic form of the solution and can be naturally extended to non-linear dimensionality reduction through the kernel trick. Experimental results on multiple real-world data sets demonstrate the effectiveness of the proposed method.

Keywords Dimensionality reduction · Supervised learning · Manifold learning · Tangent space

1 Introduction

Dimensionality reduction is a learning task that aims to find a low-dimensional representation of high-dimensional data, while preserving data information as much as possible. Processing data in the low-dimensional space can reduce computational cost and suppress noises. Provided that dimensionality reduction is performed appropriately, the discovered low-dimensional representations of data will benefit subsequent tasks, e.g., classification, clustering, data visualization.

Yang Zhou
Shanghai Key Laboratory of Multidimensional Information Processing, Department of Computer Science and Technology, East China Normal University, 500 Dongchuan Road, Shanghai 200241, P. R. China

Shiliang Sun (Corresponding Author)
Shanghai Key Laboratory of Multidimensional Information Processing, Department of Computer Science and Technology, East China Normal University, 500 Dongchuan Road, Shanghai 200241, P. R. China
Tel.: +86-21-54345183
Fax: +86-21-54345119
E-mail: slsun@cs.ecnu.edu.cn, shiliangsun@gmail.com

PCA, as an unsupervised dimensionality reduction method, seeks a set of orthogonal projection directions along which the sum of variances of data is maximized. Some other popular unsupervised methods are geometrically motivated, which aim to discover the geometrical structure of the underlying manifold, such as Laplacian Eigenmaps [2], Hessian Eigenmaps [5], Locally Linear Embedding [9], Locality Preserving Projections [7], and Local Tangent Space Alignment [17], etc. Although unsupervised approaches can reveal the underlying data manifold, they may not be the best choices for some learning scenarios because they are not able to utilize the discriminative information from data labels.

LDA is a supervised dimensionality reduction method. It finds a subspace in which data points from different classes are projected far away from each other, while those belonging to the same class are projected as close as possible. However, LDA tends to get undesirable results when data are multimodal [6] or are mainly characterized by their variances. The reason why this happens lies in the assumption adopted by LDA that data points belonging to each class are generated from the multivariate Gaussian distributions with the same covariance matrix but different means. This assumption is invalid in dealing with the data formed by several separate clusters or those living on an underlying manifold.

To solve this problem, Subclass Discriminant Analysis [18] approximates the potential distribution of data with a mixture of Gaussian distributions. More specifically, it first divides each class into a set of subclasses through clustering and then performs LDA on the divided data. Another way to overcome the drawback of LDA is preserving the data structure locally. Marginal Fisher Analysis (MFA) [15] aims to gather the nearby examples of the same class, and separate the marginal examples belonging to different classes. Locality Sensitive Discriminant Analysis (LSDA) [3] maps data points into a subspace where the examples with the same label at each local area are close, while the nearby examples from different classes are apart from each other. Local Fisher Discriminant Analysis (LFDA) [12] also focuses on discovering the local data structure. It is equivalent to operate LDA in the local scope around each example. In fact, these local structure oriented methods actually fall into the same graph Laplacian based framework. All of them employ the Laplacian matrix to preserve the local geometry of the data manifold. However, this framework fails to discover the local manifold information from tangent spaces which could be very useful and can enhance the performance of dimensionality reduction in some situations [11, 13].

In this paper, we present a novel supervised dimensionality reduction method named *local Tangent Space Discriminant analysis* (TSD). Unlike previous approaches using the graph Laplacian to discover the data manifold, our method uses the first-order Taylor expansion to represent the geometry of the local area around each data point. This strategy provides us with a natural way to utilize the information from tangent spaces. Then we seek a linear transformation to preserve the local manifold structure of the data belonging to the same class as much as possible, while maximizing the marginal data points with different class labels. As a result, the geometrical information from tangent spaces can be readily incorporated into the proposed method to improve the performance of dimensionality reduction. Moreover, the objective function of our method can be optimized analytically by solving a generalized eigenvalue problem. This also leads to a natural extension for non-linear dimensionality reduction through the kernel trick.

The rest of this paper is organized as follows. We briefly review some related work including MFA, LSDA and LFDA, and show how these methods can be considered in the same framework in Section 2. Then the *local Tangent Space Discriminant analysis* (TSD) along with its kernelization are introduced in Section 3. Section 4 discusses the connection and difference between the proposed method and related work. In Section 5, experimental results are presented. Finally, we give concluding remarks and discuss some future work in Section 6.

2 Related Work

Many dimensionality reduction methods have been proposed in recent years. Although they have different names and are derived from various motivations, a large portion of them fall into an unified graph Laplacian based framework. For example, Yan et al. [15] proposed a dimensionality reduction framework called Graph Embedding which can group together many popular dimensionality reduction approaches into a general formulation. In this section, we first introduce the Graph Embedding framework, and then briefly review some related work and show how they fall into the same framework. Finally, we provide a brief summary to discuss the strength and weakness of this framework.

2.1 Graph Embedding

Given a data set X consisting of n examples and labels, $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, where $\boldsymbol{x}_i \in \mathbb{R}^d$ denotes a d-dimensional example, $y_i \in \{1, 2, \ldots, C\}$ denotes the class label corresponding to $\boldsymbol{x}_i$, and C is the total number of classes. The relationship between each example can be easily characterized by a undirected weighted graph $G = \{X, W\}$, where each example is served as the vertex of G and a symmetric weight matrix $W \in \mathbb{R}^{n \times n}$ records the weight on the edge of each pair of vertices. W measures the similarity between each example, and its characteristic varies as the criterion of similarity changes. Generally, if two examples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are "close", the corresponding weight W_{ij} is large, whereas if they are "far away", then the W_{ij} is small. Provided a certain W, the intrinsic geometry of graph G can be represented by the Laplacian matrix [4], which is defined as

$$L = D - W, \tag{1}$$

where D is a diagonal matrix with the i-th diagonal element being $D_{ii} = \sum_{j \neq i} W_{ij}$. The Laplacian matrix is capable of representing certain geometry of data according to a specific weight matrix, and thus can be used for dimensionality reduction.

To find a good low-dimensional embedding $\boldsymbol{b} = (b_{\boldsymbol{x}_1}, b_{\boldsymbol{x}_2}, \ldots, b_{\boldsymbol{x}_n})^\top$ from high-dimensional data, we have to preserve the intrinsic geometry of the original data as much as possible. Therefore, it is natural to seek the embedding preserving the most information from G in each dimension. This graph-preserving criterion can be formulated as follows [15]:

$$\begin{aligned} \boldsymbol{b}^* &= \arg \min_{\boldsymbol{b}^\top L_b \boldsymbol{b} = a} \sum_{i \neq j} W_{ij} \|\boldsymbol{b}_i - \boldsymbol{b}_j\|^2 \\ &= \arg \min_{\boldsymbol{b}^\top L_p \boldsymbol{b} = a} \boldsymbol{b}^\top L \boldsymbol{b}, \end{aligned} \tag{2}$$

where L is the Laplacian matrix of G defined in (1), L_p is the penalty constraint matrix, and a is a constant defined to avoid a trivial solution of the objective function. Note that L_p can have multiple forms, which is usually a diagonal matrix or the Laplacian matrix of a penalty graph $G_p = \{X, W^p\}$ constructed by the same vertices X yet a different weight matrix W^p.

In this paper, we mainly focus on the linear dimensionality reduction, and the embedding of each example $\boldsymbol{x}_i$ is computed as $b_{\boldsymbol{x}_i} = \boldsymbol{t}^\top \boldsymbol{x}_i$ where $\boldsymbol{t}$ is a projection vector. In this case, (2) becomes:

$$\boldsymbol{t}^* = \arg \min_{\boldsymbol{t}^\top X L^p X^\top \boldsymbol{t} = a} = \boldsymbol{t}^\top X L X^\top \boldsymbol{t}. \tag{3}$$

The objective function (3) can be converted to a generalized eigenvalue problem:

$$X L X^\top \boldsymbol{t} = \lambda X L^p X^\top \boldsymbol{t}. \tag{4}$$

whose solution can be easily given by the eigenvector with respect to the smallest eigenvalue.

Given the above results, many local structure oriented dimensionality reduction approaches, which closely related to our proposed method, can be grouped into the unified framework. Next, we mainly discuss three of them including MFA, LSDA and LFDA.

2.2 Marginal Fisher Analysis

Marginal Fisher Analysis (MFA) is a dimensionality reduction approach that is directly derived from the Graph Embedding framework [15]. The main idea of MFA is to preserve the intraclass compactness represented by an intrinsic graph under the constraint that the interclass separability characterized by a penalty graph should be kept. For this purpose, each example is connected to its k_1-nearest neighbors belonging to the same class in the intrinsic graph, and the penalty graph is built by connecting k_2-nearest pairs of the marginal point in different classes.

For each example, let $N_{k_1}(i)$ be the set of the k_1-nearest neighbors of $\boldsymbol{x}_i$ in the same class, and $P_{k_2}(i)$ indicates the set of the k_2-nearest pairs among the set $\{(i,j), y_i \neq y_j\}$. The intraclass compactness is formulated by a local within-class scatter matrix $\bar{S}_w$:

$$\begin{aligned}\bar{S}_w =& \sum_i \sum_{i \in N_{k_1}(j) \ or \ j \in N_{k_1}(i)} (\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top \\ =& 2X(\bar{D} - \bar{W})X^\top \\ =& 2X\bar{L}X^\top,\end{aligned}$$

where $\bar{L} = \bar{D} - \bar{W}$ is the Laplacian matrix, $\bar{D}$ is a diagonal matrix with the i-th diagonal element being $\bar{D}_{ii} = \sum_{j \neq i} \bar{W}_{ij}$, and the weight matrix $\bar{W}$ is defined as follows:

$$\bar{W}_{ij} = \begin{cases} 1 & if \ i \in N_{k_1}(j) \ or \ j \in N_{k_1}(i) \\ 0 & else. \end{cases}$$

Similarly, the interclass separability can be characterized by a local between-class scatter matrix $\bar{S}_b$:

$$\begin{aligned}\bar{S}_b =& \sum_i \sum_{(i,j)\in P_{k_2}(i) \text{ or } (i,j)\in P_{k_2}(j)} (\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top \\ =& 2X(\bar{D}^p - \bar{W}^p)X^\top \\ =& 2X\bar{L}^p X^\top,\end{aligned}$$

where $\bar{L}^p = \bar{D}^p - \bar{W}^p$ is the Laplacian matrix, $\bar{D}^p$ is a diagonal matrix with the i-th diagonal element being $\bar{D}^p_{ii} = \sum_{j\neq i} \bar{W}^p_{ij}$ and the weight matrix $\bar{W}^p$ is defined as follows:

$$\bar{W}^p_{ij} = \begin{cases} 1 & if\ (i,j) \in P_{k_2}(i)\ or\ (i,j) \in P_{k_2}(j) \\ 0 & else. \end{cases}$$

Following the graph-preserving criterion presented in (2), the objective function of MFA can be written as follows:

$$\begin{aligned}\boldsymbol{t}^{MFA} =& \arg\min_{\boldsymbol{t}^\top \bar{S}_b \boldsymbol{t} = a} \boldsymbol{t}^\top \bar{S}_w \boldsymbol{t} \\ =& \arg\min_{\boldsymbol{t}^\top X\bar{L}^p X^\top \boldsymbol{t} = a} = \boldsymbol{t}^\top X\bar{L}X^\top \boldsymbol{t}.\end{aligned} \tag{5}$$

2.3 Locality Sensitive Discriminant Analysis

Another related method is Locality Sensitive Discriminant Analysis (LSDA) [3] which assumes that data live on or close to a manifold. It aims to preserve the local geometrical structure of the manifold while maximizing the local margin between different classes.

LSDA seeks a linear projection $\boldsymbol{t}$ optimizing

$$\begin{aligned}&\min \sum_{ij} \bar{W}^w_{ij} (\boldsymbol{t}^\top \boldsymbol{x}_i - \boldsymbol{t}^\top \boldsymbol{x}_j)^2, \\ &\max \sum_{ij} \bar{W}^b_{ij} (\boldsymbol{t}^\top \boldsymbol{x}_i - \boldsymbol{t}^\top \boldsymbol{x}_j)^2\end{aligned}$$

under the constraint that $\boldsymbol{t}^\top X\bar{D}^w X^\top \boldsymbol{t} = 1$. Let $N^w(i)$ be the set of the k-nearest neighbors of $\boldsymbol{x}_i$ sharing the same label y_i, and $N^b(i)$ be the set of the k-nearest neighbors of $\boldsymbol{x}_i$ having the labels different from y_i. Then the weight matrices $\bar{W}^w$ and $\bar{W}^b$ are defined as:

$$\begin{aligned}\bar{W}^w_{ij} =& \begin{cases} 1 & if\ i \in N^w(j)\ or\ j \in N^w(i) \\ 0 & else, \end{cases} \\ \bar{W}^b_{ij} =& \begin{cases} 1 & if\ i \in N^b(j)\ or\ j \in N^b(i) \\ 0 & else. \end{cases}\end{aligned}$$

The objective function of LSDA described above can be formulated as follows:

$$\begin{aligned}\boldsymbol{t}^{LSDA} =& \arg\max_{\boldsymbol{t}^\top X\bar{D}^w X^\top \boldsymbol{t} = 1} \boldsymbol{t}^\top X(\alpha\bar{L}^b + (1-\alpha)\bar{W}^w)X^\top \boldsymbol{t} \\ =& \arg\min_{\boldsymbol{t}^\top X\bar{D}^w X^\top \boldsymbol{t} = 1} \boldsymbol{t}^\top X((\alpha-1)\bar{W}^w - \alpha\bar{L}^b)X^\top \boldsymbol{t},\end{aligned} \tag{6}$$

where α is a trade-off parameter, $\bar{L}^b$ is the Laplacian matrix constructed by $\bar{W}^b$, and $\bar{D}^w$ is a diagonal matrix with the i-th diagonal element being $\bar{D}^w_{ii} = \sum_{j \neq i} \bar{W}^w_{ij}$. LSDA follows the framework defined in (3) and the solution $\boldsymbol{t}^{LSDA}$ s given by (4) with $L = (\alpha - 1)\bar{W}^w - \alpha\bar{L}^b$ and $L^p = \bar{D}^w$.

2.4 Local Fisher Discriminant Analysis

Local Fisher Discriminant Analysis (LFDA) [12] combines the ideas of LDA and LPP [7] to overcome the problem that LDA [6] can not appropriately handle the data with multimodality. More specifically, it evaluates the levels of the between-class scatter and the within-class scatter in a local manner, and tries to attain the local between-class separation and the local within-class structure preservation at the same time [12].

Let $\tilde{S}_w$ and $\tilde{S}_b$ be the local within-class scatter matrix and the local between-class scatter matrix defined by

$$\begin{aligned}\tilde{S}_w =& \frac{1}{2}\sum_{ij} \tilde{W}^w_{ij}(\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top = X\tilde{L}^w X^\top,\\ \tilde{S}_b =& \frac{1}{2}\sum_{ij} \tilde{W}^b_{ij}(\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top = X\tilde{L}^b X^\top,\end{aligned}$$

where $\tilde{L}^w$ and $\tilde{L}^b$ are the Laplacian matrices constructed by the weight matrices $\tilde{W}^w_{ij}$ and $\tilde{W}^b_{ij}$ with

$$\begin{aligned}\tilde{W}^w_{ij} =& \begin{cases} A_{ij}/n_c & if\ y_i = y_j \\ 0 & if\ y_i \neq y_j, \end{cases}\\ \tilde{W}^b_{ij} =& \begin{cases} A_{ij}(1/n - 1/n_c) & if\ y_i = y_j \\ 1/n & if\ y_i \neq y_j. \end{cases}\end{aligned}$$

n_c denotes the number of examples from the c-th class, and A_{ij} is a weight that indicates the similarity between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, whose definition is given as follows:

$$A_{ij} = \begin{cases} \exp(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{\sigma_i \sigma_j}) & if\ i \in N_k(j)\ or\ j \in N_k(i) \\ 0 & else, \end{cases}$$

where σ_i is set to be the distance between $\boldsymbol{x}_i$ and its k-th nearest neighbor.

LFDA seeks a linear projection so that $\tilde{S}_w$ is minimized and $\tilde{S}_b$ is maximized. Essentially, this strategy is equivalent to find a projection which fits the Fisher criterion in the local area around each example. The optimization problem of LFDA is given as follows:

$$\begin{aligned}\boldsymbol{t}^{LFDA} =& \arg\max_{\boldsymbol{t}} \frac{\boldsymbol{t}^\top \tilde{S}_b \boldsymbol{t}}{\boldsymbol{t}^\top \tilde{S}_w \boldsymbol{t}} = \arg\max_{\boldsymbol{t}} \frac{\boldsymbol{t}^\top X\tilde{L}^w X^\top \boldsymbol{t}}{\boldsymbol{t}^\top X\tilde{L}^b X^\top \boldsymbol{t}}\\ =& \arg\min_{\boldsymbol{t}^\top X\tilde{L}^b X^\top \boldsymbol{t} = 1} = \boldsymbol{t}^\top X\tilde{L}X^\top \boldsymbol{t}. \end{aligned} \tag{7}$$

According to (7), it is easy to find that LFDA also falls into the Graph Embedding framework defined in (3) with $L = \tilde{L}^w$, $L^p = \tilde{L}^b$ and $a = 1$.

2.5 A Brief Summary

With the above results, it is easy to find that all methods mentioned above can be considered in the same graph Laplacian based framework and the main difference among them only lies in the different graphs adopted by each method. For different methods, such graphs are constructed by certain weight matrices to incorporate specific neighborhood information of the data set. One important merit of this framework is that it not only takes advantage of the facility of the Laplacian matrix to preserve the local geometry of the data manifold, but also benefits from the elegant formulation which can be easily optimized through the generalized eigenvalue decomposition.

Although the graph Laplacian provides us with a powerful and flexible tool to discover the underlying data manifold, it fails to discover the local geometrical structure from tangent spaces, and thus may lose much useful information whose effectiveness has been shown in many applications especially in the handwritten digit recognition [11]. Moreover, the graph Laplacian may fail to capture meaningful manifold structures, when data are sparsely distributed in the original space. In this case, the graph Laplacian constructed by sparsely distributed data in the high-dimensional space may not be able to discover the correct underlying manifold, since it can hardly connect sparse data points into a smooth manifold. On the other hand, the tangent spaces of the underlying manifold, which are low-dimensional in nature, can reflect the manifold structure in each local area. This implies that tangent spaces are very useful for learning the data manifold. Then how to develop a dimensionality reduction algorithm which is capable of combining the flexibility of the graph Laplacian with the utility of tangent spaces? To solve this problem, we present our algorithm which can readily use the structural information from tangent spaces for supervised dimensionality reduction.

3 Local Tangent Space Discriminant Analysis

In this section, we present the *local Tangent Space Discriminant analysis* (TSD) algorithm and its non-linear extension. As a supervised dimensionality reduction method, TSD aims to seek an embedding space where the local manifold structure of the data belonging to the same class is preserved as much as possible, and the marginal data points with different class labels are better separated. Compared with the methods discussed in Section 2, the key advantage of our algorithm is that it is capable of capturing the local manifold structure from tangent spaces without losing the analytic form of the solution.

3.1 Preliminaries

To begin with, we briefly introduce the concepts of the tangent space and tangent vector. In differential geometry, one can attach to every point $\boldsymbol{x}$ of a differentiable manifold $\mathcal{M}$ a tangent space $T_{\boldsymbol{x}}\mathcal{M}$ in which every vector tangentially passes through $\boldsymbol{x}$. The elements of the tangent space are called tangent vectors at $\boldsymbol{x}$, which is a vector that is tangent to a curve or surface at $\boldsymbol{x}$ (see Fig. 1 for the

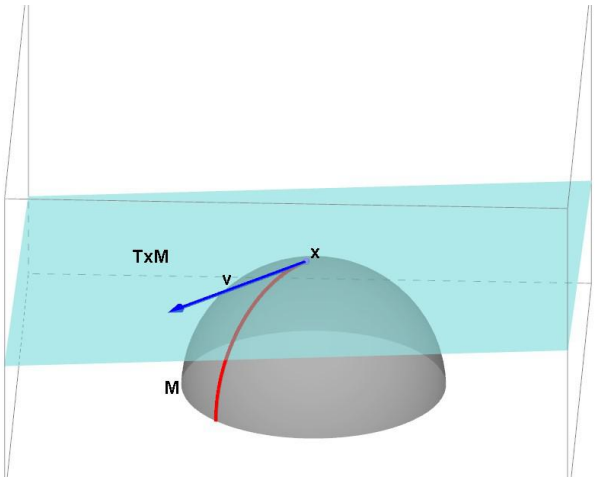

Fig. 1 The tangent space $T_{\boldsymbol{x}}\mathcal{M}$ and a tangent vector $\boldsymbol{v} \in T_{\boldsymbol{x}}\mathcal{M}$, along a curve travelling through $\boldsymbol{x} \in \mathcal{M}$.

illustration). All the tangent spaces of a connected manifold have the same dimension, equal to the dimension of the manifold. In practice, if the manifold is smooth enough, the subspace constructed by performing PCA on the neighborhood of $\boldsymbol{x}$ can be a good approximation of the tangent space at $\boldsymbol{x}$ [14], since the nearby data points of $\boldsymbol{x}$ can be viewed as approximately lying in a subspace which is tangent to the data manifold. Once tangent spaces and tangent vectors have been introduced, they can serve to characterize a differentiable curve on the manifold whose derivative at any point is equal to the tangent vector attached to that point. This is a crucial property that plays a key role in deriving our TSD algorithm.

In recent years, some tangent space based dimensionality reduction methods have been proposed by using the above property. They learn the data manifold by estimating a function whose value can serve as a low-dimensional representation of the manifold. Tangent Space Intrinsic Manifold Regularization (TSIMR) [13] estimates a local linear function on the manifold which has constant manifold derivatives. Parallel Vector Field Embedding (PFE) [8] represents a function along the manifold from the perspective of vector fields and requires the vector field at each data point to be as parallel as possible. Although they are effective to preserve the manifold geometry, these tangent space based methods are unsupervised in nature, which have no ability to utilize the discriminant information of class labels. Therefore, they are not optimal for the supervised case. To solve this problem, we propose the TSD algorithm which partly shares the same spirit with TSIMR and PFE but is optimal for supervised dimensionality reduction.

3.2 The Algorithm

Suppose that data are sampled from an m-dimensional smooth manifold $\mathcal{M}$ in a d-dimensional space. Let $\mathcal{T}_{\boldsymbol{z}}\mathcal{M}$ denotes the tangent space attached to $\boldsymbol{z}$, where $\boldsymbol{z} \in \mathcal{M}$ is a fixed data point on the $\mathcal{M}$. Motivated by Tangent Space Intrinsic Manifold Regularization (TSIMR) [13], we represent the local manifold structure of data by means of tangent spaces. According to the first-order Taylor expansion at $\boldsymbol{z}$, any function f defined on the manifold $\mathcal{M}$ can be expressed as:

$$f(\boldsymbol{x}) = f(\boldsymbol{z}) + \boldsymbol{w_z}^\top \boldsymbol{u_z}(\boldsymbol{x}) + O(\|\boldsymbol{x} - \boldsymbol{z}\|^2),$$

where $\boldsymbol{x} \in \mathbb{R}^d$ is a d-dimensional data point and $\boldsymbol{u_z}(\boldsymbol{x}) = T_{\boldsymbol{z}}^\top(\boldsymbol{x} - \boldsymbol{z})$ is an m-dimensional tangent vector which gives the m-dimensional representation of $\boldsymbol{x}$ in $\mathcal{T}_{\boldsymbol{z}}\mathcal{M}$. $T_{\boldsymbol{z}}$ is a $d \times m$ matrix formed by the orthonormal bases of $\mathcal{T}_{\boldsymbol{z}}\mathcal{M}$, which can be estimated through local PCA, i.e., performing standard PCA on the neighborhood of $\boldsymbol{z}$. And $\boldsymbol{w_z}$ is an m-dimensional vector representing the directional derivative of f at $\boldsymbol{z}$ with respect to $\boldsymbol{u_z}(\boldsymbol{x})$ on the manifold $\mathcal{M}$.

In the scenario of dimensionality reduction, $f(\boldsymbol{x})$ denotes a one-dimensional embedding of $\boldsymbol{x}$. If there are two data points $\boldsymbol{z}$ and $\boldsymbol{z}'$ have a small Euclidean distance, by using the first-order Taylor expansion at $\boldsymbol{z}'$, the embedding $f(\boldsymbol{z})$ can be represented as:

$$f(\boldsymbol{z}) = f(\boldsymbol{z}') + \boldsymbol{w}_{\boldsymbol{z}'}^\top T_{\boldsymbol{z}'}^\top(\boldsymbol{z} - \boldsymbol{z}') + O(\|\boldsymbol{z} - \boldsymbol{z}'\|^2). \tag{8}$$

Suppose that the data can be well characterized by a linear function on the underlying manifold $\mathcal{M}$. Then we can omit the remainders in (8) because the second-order derivatives of f vanishes. Therefore, provided $\boldsymbol{z}$ and $\boldsymbol{z}'$ are close enough, any embedding $f(\boldsymbol{z})$ can be well approximated by a linear function as follows:

$$f(\boldsymbol{z}) \approx f(\boldsymbol{z}') + \boldsymbol{w}_{\boldsymbol{z}'}^\top T_{\boldsymbol{z}'}^\top(\boldsymbol{z} - \boldsymbol{z}'). \tag{9}$$

Based on the above results, we know that every data point in a local area should satisfies (9), which leads to a natural criterion of preserving the local manifold structure of data. Suppose that the training data include n examples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ belonging to C classes where $\boldsymbol{x}_i \in \mathbb{R}^d$ is a d-dimensional example, and $y_i \in \{1, 2, \ldots, C\}$ is the class label associated with the example $\boldsymbol{x}_i$. Consider a linear projection $\boldsymbol{t} \in \mathbb{R}^d$ which maps the data to a one-dimensional embedding. Then the embedding of $\boldsymbol{x}$ can be expressed as $f(\boldsymbol{x}) = \boldsymbol{t}^\top \boldsymbol{x}$. We aim to find a projection $\boldsymbol{t}$ to minimize the difference between the l.h.s and the r.h.s of (9) for every example and its neighbors belonging to the same class, and to better separate the marginal data points in different classes.

In order to minimize the difference between the l.h.s and the r.h.s of (9) for nearby intraclass data, we need to construct the within-class graph G^w. For the within-class graph G^w, if $\boldsymbol{x}_i$ is among the k_1-nearest neighbors of $\boldsymbol{x}_j$ with $y_i = y_j$, an edge is added between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, and the elements of the weight matrix W^w are set to $W_{ij}^w = W_{ji}^w = 1$. Then we can formulate a within-class objective function as follows:

$$\min \sum_{ij} W_{ij}^w (\boldsymbol{t}^\top \boldsymbol{x}_i - \boldsymbol{t}^\top \boldsymbol{x}_j - \boldsymbol{w}_{\boldsymbol{x}_j}^\top T_{\boldsymbol{x}_j}^\top (\boldsymbol{x}_i - \boldsymbol{x}_j))^2, \tag{10}$$

$$W_{ij}^w = \begin{cases} 1 & if\ i \in N_{k_1}(j)\ or\ j \in N_{k_1}(i) \\ 0 & else, \end{cases} \tag{11}$$

where $N_{k_1}(i)$ denotes the set of the k-nearest neighbors of $\boldsymbol{x}_i$ sharing the same label y_i, and the orthonormal base matrix $T_{\boldsymbol{x}_i}$ of the tangent space $\mathcal{T}_{\boldsymbol{x}_i}\mathcal{M}$ at each data point $\boldsymbol{x}_i$ are computed by performing PCA on the k_1-nearest neighborhood of $\boldsymbol{x}_i$.

To separate the marginal data points in different classes, we need to maximize and the distances of the embeddings of nearby interclass data points. To this end, we need to construct the between-class graph G^b. For the between-class graph G^b, if the pair (i,j) is among the k_2-shortest pairs in the set $\{(i,j), y_i \neq y_j\}$, an edge is added between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, and the elements of the weight matrix W^b are set to $W^b_{ij} = 1$. Similar to the above objective function, we can write a between-class objective function as follows:

$$\max \sum_{ij} W^b_{ij} (\boldsymbol{t}^\top \boldsymbol{x}_i - \boldsymbol{t}^\top \boldsymbol{x}_j)^2, \tag{12}$$

$$W^b_{ij} = \begin{cases} 1 & if\ (i,j) \in P_{k_2}(i)\ or\ (i,j) \in P_{k_2}(j) \\ 0 & else, \end{cases} \tag{13}$$

where $P_{k_2}(i)$ indicates the set of the k_2-nearest pairs among the set $\{(i,j), y_i \neq y_j\}$.

Note that the terms $\boldsymbol{w}_{\boldsymbol{x}_j}^\top T_{\boldsymbol{x}_j}^\top (\boldsymbol{x}_i - \boldsymbol{x}_j)$ $(i,j = 1,\ldots,n)$ in (10) distinguish our strategy of preserving the local data structure from the graph Laplacian based one, where $\boldsymbol{w}_{\boldsymbol{x}_j}$ is a coefficient vector and should be optimized with $\boldsymbol{t}$ simultaneously. These terms characterize how well two different examples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ fit into the local linear approximation of f, which leads to an appropriate way to preserve the local intraclass geometry along the manifold $\mathcal{M}$. Therefore, our strategy can extract more geometrical information from the data than the graph Laplacian based one. Moreover, any valid weight matrix W^w, such as the one used in LFDA, can be used to preserve specific geometrical structure of the data manifold. This free-form property of the weight matrix is of great importance when we want to apply dimensionality reduction to various types of data.

The objective function (10) can be reformulated as a canonical matrix quadratic form as follows:

$$\begin{aligned} &\sum_{ij} W^w_{ij} (\boldsymbol{t}^\top \boldsymbol{x}_i - \boldsymbol{t}^\top \boldsymbol{x}_j - \boldsymbol{w}_{\boldsymbol{x}_j}^\top T_{\boldsymbol{x}_j}^\top (\boldsymbol{x}_i - \boldsymbol{x}_j))^2 \\ =&\begin{pmatrix} \boldsymbol{t} \\ \boldsymbol{w} \end{pmatrix}^\top \begin{pmatrix} XS_1X^\top & XS_2 \\ {S_2}^\top X^\top & S_3 \end{pmatrix} \begin{pmatrix} \boldsymbol{t} \\ \boldsymbol{w} \end{pmatrix} \\ =&\begin{pmatrix} \boldsymbol{t} \\ \boldsymbol{w} \end{pmatrix}^\top S \begin{pmatrix} \boldsymbol{t} \\ \boldsymbol{w} \end{pmatrix}, \end{aligned} \tag{14}$$

where we have defined $\boldsymbol{w} = (\boldsymbol{w}_{\boldsymbol{x}_1}^\top, \boldsymbol{w}_{\boldsymbol{x}_2}^\top, \ldots, \boldsymbol{w}_{\boldsymbol{x}_n}^\top)^\top$, $X = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ is the data matrix, and S is a $(d+mn) \times (d+mn)$ positive semi-definite matrix constructed by four blocks, i.e., $XS_1X^\top$, XS_2, ${S_2}^\top X^\top$ and S_3. For simplicity, we omit the detailed derivation of S here, which is available in the Appendix A.

Recall that $\boldsymbol{w}_{\boldsymbol{x}_i}$ is the directional derivative of f at $\boldsymbol{x}_i$. Note that the linear projection vector $\boldsymbol{t}$ is under the influences of both the direction and the length of each $\boldsymbol{w}_{\boldsymbol{x}_i}$. To make within-class examples further compacted, we hope that the projection $\boldsymbol{t}$ is more effected by $\boldsymbol{w}_{\boldsymbol{x}_i}$'s direction than its length. Therefore, it makes sense to regularize the length of $\boldsymbol{w}_{\boldsymbol{x}_i}$ $(i = 1,\ldots,n)$. This can be achieved by adding

a regularizer $\|\boldsymbol{t}\|^2 + \sum_i \|\boldsymbol{w}_{\boldsymbol{x}_i}\|^2$ to (14). Define $\boldsymbol{f} = (\boldsymbol{t}^\top, \boldsymbol{w}^\top)^\top$, the optimization function turns out to be:

$$\begin{aligned}&\boldsymbol{f}^\top S\boldsymbol{f} + \gamma(\|\boldsymbol{t}\|^2 + \sum_i \|\boldsymbol{w}_{\boldsymbol{x}_i}\|^2)\\ =&\boldsymbol{f}^\top S\boldsymbol{f} + \gamma\|\boldsymbol{f}\|^2 = \boldsymbol{f}^\top(S+\gamma I)\boldsymbol{f},\end{aligned} \tag{15}$$

where $\gamma > 0$ is a trade-off parameter. In fact, the extra term $\gamma\|\boldsymbol{f}\|^2$ often refers to as the Tikhonov regularizer, which is commonly used with a small γ to keep matrices from being singular. However, the value of γ is crucial for our method, because it also controls the influence of $\boldsymbol{w}$'s length on the projection $\boldsymbol{t}$.

With simple algebraic formulation, the objective function (12) becomes

$$\begin{aligned}&\sum_{ij} W^b_{ij}(\boldsymbol{t}^\top\boldsymbol{x}_i - \boldsymbol{t}^\top\boldsymbol{x}_j)^2\\ =&2\boldsymbol{t}^\top X(D^b - W^b)X^\top\boldsymbol{t} = 2\boldsymbol{t}^\top XL^bX^\top\boldsymbol{t}\\ =&\begin{pmatrix}\boldsymbol{t}\\ \boldsymbol{w}\end{pmatrix}^\top\begin{pmatrix}2XL^bX^\top & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{pmatrix}\begin{pmatrix}\boldsymbol{t}\\ \boldsymbol{w}\end{pmatrix}\\ =&\begin{pmatrix}\boldsymbol{t}\\ \boldsymbol{w}\end{pmatrix}^\top S_b\begin{pmatrix}\boldsymbol{t}\\ \boldsymbol{w}\end{pmatrix},\end{aligned} \tag{16}$$

where $L^b = D^b - W^b$ is the Laplacian matrix, and D^b is a diagonal matrix with the i-th diagonal element being $D^b_{ii} = \sum_{j\neq i} W^b_{ij}$.

Finally, by integrating (15) and (16), the objective function of TSD can be written as follows:

$$\boldsymbol{f}^* = \arg\max_{\boldsymbol{f}} \frac{\boldsymbol{f}^\top S_b\boldsymbol{f}}{\boldsymbol{f}^\top(S+\gamma I)\boldsymbol{f}}. \tag{17}$$

The optimization of (17) can be achieved by solving a generalized eigenvalue problem:

$$S_b\boldsymbol{f} = \lambda(S+\gamma I)\boldsymbol{f} \tag{18}$$

whose solution can be easily given by the eigenvector with respect to the largest eigenvalue. In order to obtain a one-dimensional embedding of an example $\boldsymbol{x}$, we just use the first part of $\boldsymbol{f}^* = (\boldsymbol{t}^{*\top}, \boldsymbol{w}^{*\top})^\top$ and compute $b = \boldsymbol{t}^{*\top}\boldsymbol{x}$. Suppose that we want to project d-dimensional data into an r-dimensional subspace. Let $\boldsymbol{f}_1, \ldots, \boldsymbol{f}_r$ be the solutions of (17) corresponding to the r largest eigenvalues $\lambda_1 > \ldots > \lambda_r$. Then the r-dimensional embedding $\boldsymbol{b}$ of $\boldsymbol{x}$ is computed as follows:

$$\boldsymbol{b} = T^\top\boldsymbol{x}, \quad T = (\boldsymbol{t}_1, \ldots, \boldsymbol{t}_r).$$

Algorithm 1 gives the pseudo-code for TSD. It is worth noting that although $\boldsymbol{w}^*$ seems not to be used in computing the low-dimensional embeddings, as the parameter which is simultaneously optimized with $\boldsymbol{t}^*$, it exerts a crucial influence on the resultant transformation matrix T. This is means that both $\boldsymbol{t}^*$ and $\boldsymbol{w}^*$ determine the final results of TSD.

The main computational cost of TSD lies in building tangent spaces for n data points and solving the generalized eigenvalue problem (18). Our algorithm has a time complexity of $O((d^2m + m^2d) \times n)$ for the construction of n tangent spaces and $O(r^2 \times (d + mn))$ for finding r eigenvectors with respect to the r

Algorithm 1 TSD

Input: Labeled examples $\{\{\boldsymbol{x}_i, y_i\}|\boldsymbol{x}_i \in \mathbb{R}^d, y_i \in \{1, 2, \ldots, C\}\}_{i=1}^n$;
Dimensionality of embedding space m $(1 \leq m \leq d)$;
Trade-off parameters γ $(\gamma > 0)$.
Output: $d \times r$ transformation matrix T.

Construct the within-class graph G^w and the between-class graph G^b;
Calculate the weight matrices W^w and W^b with (11) and (13);
for $i = 1$ **to** n **do**
  Construct $T_{\boldsymbol{x}_i}$ by performing PCA on the intraclass neighborhood of $\boldsymbol{x}_i$;
end for
Compute the eigenvectors $\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_r$ of (18) with respect to the top r eigenvalues;
$T = (\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots, \boldsymbol{t}_r)$.

largest eigenvalues. For comparison, we also give the time complexities of some classical and related methods. The time complexity of PCA is $O(n^2d)$ and that of LDA is also $O(n^2d)$ [16]. As we have discussed in Section 2, MFA, LSDA, LFDA fall into the same framework with different graphs, which implies that they have the same computational cost. Since LDA also falls into the graph Laplacian based framework [15], their time complexities turn out to be $O(n^2d)$. The above analysis suggests that TSD is more time consuming compared with other methods. However, since local tangent spaces are estimated by local PCA, we can obtain at most $k_1 + 1$ meaningful orthonormal bases for each tangent space[1], where k_1 is the size of within-class neighborhood. This implies that the dimensionality m of the directional derivative $\boldsymbol{w}_{\boldsymbol{x}_i}$ $(i = 1, \ldots, n)$ is always less than $k_1 + 1$. In practice, k_1 is usually small to ensure the locality. This makes sure that m is actually a small constant. To conclude, the overall time complexity of TSD is $O((d^2m + m^2d) \times n + r^2 \times (d + mn))$. Since m is usually small, TSD has an acceptable computational cost.

3.3 Kernel TSD

TSD is a linear dimensionality reduction method. In this section, we propose Kernel TSD which can be performed in a Reproducing Kernel Hilbert Space (RKHS) for non-linear dimensionality reduction.

Consider a feature space $\mathcal{F}$ induced by a non-linear mapping $\phi : \mathcal{X} \to \mathcal{F}$, where $\mathcal{X}$ is an input domain. We can construct an RKHS $H_{\mathcal{K}}$ by defining a kernel function $\mathcal{K}(\cdot, \cdot)$ using the inner product operation $\langle \cdot, \cdot \rangle$, such that $\mathcal{K}(\boldsymbol{x}, \boldsymbol{y}) = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{y}) \rangle$. Given a data set $\{\boldsymbol{x}_i \in \mathcal{X}\}_{i=1}^n$, we can define the data matrix in the feature space $\mathcal{F}$ as $\boldsymbol{\Phi} = (\phi(\boldsymbol{x}_1), \ldots, \phi(\boldsymbol{x}_n))$. Then one can use the orthogonal projection to decompose any projection vector $\boldsymbol{t} \in H_{\mathcal{K}}$ into a sum of two functions: one lying in the $span\{\phi(\boldsymbol{x}_1), \ldots, \phi(\boldsymbol{x}_n)\}$, and the other lying in the orthogonal complementary space. Therefore, there exist a set of coefficients α_i $(i = 1, 2, \ldots, n)$ satisfying

$$\boldsymbol{t} = \sum_{i=1}^{n} \alpha_i \phi(\boldsymbol{x}_i) + \boldsymbol{v} = \boldsymbol{\Phi}\boldsymbol{\alpha} + \boldsymbol{v}, \tag{19}$$

[1] That's because there are only $k_1 + 1$ examples as the inputs of local PCA.

where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_n)^\top$ and $\langle \boldsymbol{v}, \phi(\boldsymbol{x}_i) \rangle = 0$ for all i. Note that although we set $\boldsymbol{f} = (\boldsymbol{t}^\top, \boldsymbol{w}^\top)^\top$ and optimize $\boldsymbol{t}$ and $\boldsymbol{w}$ together, we should estimate tangent spaces in $\mathcal{F}$ through local Kernel PCA [10] rather than reparametrize $\boldsymbol{w}$ like $\boldsymbol{t}$.

Let $T^\phi_{\boldsymbol{x}_i}$ be the matrix formed by the orthonormal bases of the tangent space attached to $\phi(\boldsymbol{x}_i)$. By replacing $T_{\boldsymbol{x}_i}$ with $T^\phi_{\boldsymbol{x}_i}$ $(i = 1, 2, \ldots, n)$ and substituting (19) into (15), the objective function (15) turns out to be:

$$\begin{aligned}
&\begin{pmatrix} \boldsymbol{t} \\ \boldsymbol{w} \end{pmatrix}^\top \begin{pmatrix} XS_1X^\top & XS_2 \\ S_2{}^\top X^\top & S_3 \end{pmatrix} \begin{pmatrix} \boldsymbol{t} \\ \boldsymbol{w} \end{pmatrix} + \gamma \begin{pmatrix} \boldsymbol{t} \\ \boldsymbol{w} \end{pmatrix}^\top \begin{pmatrix} \boldsymbol{t} \\ \boldsymbol{w} \end{pmatrix} \\
=&\begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{w} \end{pmatrix}^\top \begin{pmatrix} KS_1K & KS_2 \\ S_2^\top K & S_3 \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{w} \end{pmatrix} + \gamma \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{w} \end{pmatrix}^\top \begin{pmatrix} K & \mathbf{0} \\ \mathbf{0} & \bar{I} \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{w} \end{pmatrix} \\
=&\begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{w} \end{pmatrix}^\top \begin{pmatrix} KS_1K + \gamma K & KS_2 \\ S_2^\top K & S_3 + \gamma \bar{I} \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{w} \end{pmatrix} \\
=&\begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{w} \end{pmatrix}^\top S^\phi \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{w} \end{pmatrix},
\end{aligned}$$

where K is a kernel matrix with $K_{ij} = \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and $\bar{I}$ is an identity matrix sized $mn \times mn$. Similarly, the objective function (16) becomes

$$\begin{aligned}
&\begin{pmatrix} \boldsymbol{t} \\ \boldsymbol{w} \end{pmatrix}^\top \begin{pmatrix} 2XL^bX^\top & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{t} \\ \boldsymbol{w} \end{pmatrix} \\
=&\begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{w} \end{pmatrix}^\top \begin{pmatrix} 2KL^bK & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{w} \end{pmatrix} \\
=&\begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{w} \end{pmatrix}^\top S_b^\phi \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{w} \end{pmatrix}.
\end{aligned}$$

Note that due to $\langle \boldsymbol{v}, \phi(\boldsymbol{x}_i) \rangle = 0$ for all i, every term of $\boldsymbol{v}$ vanishes from the above formulations. Finally, Kernel TSD can be converted to a generalized eigenvalue problem as follows:

$$S_b^\phi \boldsymbol{\varphi} = \lambda S^\phi \boldsymbol{\varphi}, \tag{20}$$

where we have defined $\boldsymbol{\varphi} = (\boldsymbol{\alpha}^\top, \boldsymbol{w}^\top)^\top$.

Given the eigenvectors $\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_r$ with respect to the r largest eigenvalues of (20), the resultant transformation matrix can be written as $\Gamma = (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_r)$. Then, the embedding $\boldsymbol{b}$ of an original example $\boldsymbol{x}$ is computed as:

$$\boldsymbol{b} = \Gamma^\top \Phi^\top \phi(\boldsymbol{x}) = \Gamma^\top (\mathcal{K}(\boldsymbol{x}_1, \boldsymbol{x}), \ldots, \mathcal{K}(\boldsymbol{x}_n, \boldsymbol{x}))^\top.$$

4 Discussion

For developing a good graph-based dimensionality reduction method, one of the most important problems is how to construct a good graph so that the preferred data structure can be preserved. As we have discussed in Section 2, many existing methods such as MFA, LSDA and LFDA aim to design specific graphs to enhance the local compactness of the data in the same class and separate the data points with different class labels. However, none of them breaks the graph Laplacian based

framework. Our method mainly focuses on developing a new strategy to extract more information of the data manifold from a given graph. More specifically, we use the first-order Taylor expansion to incorporate the structural information from tangent spaces, i.e., the terms $\boldsymbol{w}_{\boldsymbol{x}_j}^\top T_{\boldsymbol{x}_j}^\top (\boldsymbol{x}_i - \boldsymbol{x}_j))^2$ $(i, j = 1, \ldots, n)$ in (10), into the scatter matrix S. Moreover, it is worth noting that although we specify a certain weight matrix (11) to construct the within-class graph G^w, our method is flexible enough to utilize the information from any valid graphs such as the one used in LFDA.

Local Tangent Space Alignment (LTSA) [17] is a popular dimensionality reduction method which also use the information from tangent spaces. The main idea of LTSA is to align every local tangent space to construct global coordinates. Although both TSD and LTSA utilize local tangent spaces, there are mainly two differences between them: 1) they actually solve different problems in essence. TSD is a linear supervised dimensionality reduction method, while LTSA is a non-linear unsupervised one. As a result, our method not only considers the class labels to make use of discriminant information, but can obtain an explicit transformation matrix to compute the mappings for out-of-sample data. 2) TSD is a graph-based method which can adopt any valid graph for training, whereas LTSA is not. This property provides TSD with much more flexibility to handle various types of data for different applications.

Our method shares the same spirit with TSIMR [13], and both of them employ tangent spaces to discover the geometrical structure of the data manifold. However, our approach and TSIMR differ in two key aspects: 1) Like LTSA, TSIMR is a non-linear unsupervised method, and thus has no ability to capture the discriminant information or give an explicit transformation matrix. 2) They have totally different objective functions. It should be noted that TSD employs (10) to construct the scatter matrix S, while the objective function of TSIMR has other terms $\|\boldsymbol{w}_{\boldsymbol{x}_i} - T_{\boldsymbol{x}_i}^\top T_{\boldsymbol{x}_j} \boldsymbol{w}_{\boldsymbol{x}_j}\|_2^2$ $(i, j = 1, \ldots, n)$. And we find these terms are not much beneficial for discriminant analysis.

The effect of the Tikhonov regularizer $\gamma\|\boldsymbol{f}\|^2 = \gamma(\|\boldsymbol{t}\|^2 + \sum_i \|\boldsymbol{w}_{\boldsymbol{x}_i}\|^2)$ in (15) should be highlighted, since it plays a key role in our method. Generally, Tikhonov regularization is a common technique employed by many dimensionality reduction methods to deal with the singularity problem of the matrix, where the parameter γ is always set to a very small value. However, our method needs an appropriate large γ to penalize large $\|\boldsymbol{w}_{\boldsymbol{x}_i}\|$ $(i = 1, \ldots, n)$ so that the within-class compactness can be enhanced.

5 Experiments

To evaluate the proposed method, related dimensionality reduction methods including PCA, LDA, MFA, LSDA and LFDA are compared with TSD on multiple real-world data sets from the UCI Machine Learning Repository [1], the Protein Sequence data set[2] from glycosylation database Uniprot (v8.0), and the USPS data set. Specifically, we first perform dimensionality reduction to map all examples into a subspace, and then carry out classification using the nearest neighbor classifier (1-NN) in the subspace. This experimental setting is the same as the one

[2] This Protein Sequence data set is available at `http://www.ebi.ac.uk/uniprot`.

Table 1 List of the classification data sets used in our experiments.

Data Set	Dimensionality	# of examples	# of classes	Rates of training
Satellite	36	6435	2	10%
Theorem Prove	51	3059	2	10%
Breast Cancer	9	263	2	50%
Column2C	6	310	2	50%
Image	18	2086	2	10%
Ionosphere	34	351	2	50%
Protein Sequence	420	2000	2	20%
Semeion Handwritten	256	1593	10	20%
USPS	256	2007	10	20%

adopted in [12]. Moreover, we also compare the baseline method that just employs the 1-NN classifier in the original space without performing dimensionality reduction.

Seven UCI data sets (Satellite, Theorem Prove, Breast Cancer, Column2C Image, Ionosphere, Semeion Handwritten), the Protein Sequence data set, and the USPS data set are used to conduct our experiments. Originally, the Theorem Prove and USPS data sets are divided into a training set and a test set. For simplicity, we just use their test sets to carry on the experiments. For the Protein Sequence data set, we use a subset of the Uniprot database which contains only 99 mammalian protein entries. For each data set, we randomly split certain rates of the data as the training set and the rest as the test set. Furthermore, all the parameters for MFA, LSDA, LFDA and TSD are selected by three-fold cross-validation. The configuration of each data set is shown in Table 1.

The performance of PCA and graph-based dimensionality methods including MFA, LSDA, LFDA and TSD depend on the dimensionality of the discovered embedding subspace. Thus we show the best results obtained by those methods. Every experimental result is obtained from the average over 20 splits. We give the mean values and standard deviations of the error rates (%) on the employed data sets, where the best method is highlighted in bold font and the best and comparable ones based on the t-test with the significance level 5% are marked with '$\triangle$'.

The experimental results on the Satellite, Theorem Prove, Breast Cancer and Column2C data sets are shown in Table 2. In most cases, classification with dimensionality reduction is statistically better than the baseline. However, LDA perform well on none of the four data sets, probably because the implicit assumption adopted by LDA mismatches the distributions of these data sets. On the other hand, all the graph-based methods get reasonable well results, because they aim to preserve the local structure of data. PCA also works well for the purpose of separating data from different classes. Although it does not attain the best performance, our method achieves comparable good results.

Table 3 describes the classification performance of each method on the Image, Ionosphere, Protein Sequence, Semeion Handwritten and USPS data sets. Again, LDA gets worse results. Surprisingly, the counterparts of TSD including MFA, LSDA, LFDA, fail to perform well for the Protein Sequence and the Semeion Handwritten data sets. In the case of Semeion Handwritten data set, these methods are even worse than the baseline. The characteristics of the feature vectors in the

Table 2 Mean values and standard deviations of the error rates (%) on the Satellite, Theorem Prove, Breast Cancer and Column2C data sets. The best method is highlighted by bold font. The best and comparable ones based on the t-test with the significance level 5% are marked with '△'.

Methods	Satellite	Theorem Prove	Breast Cancer	Column2C
Baseline	13.31±0.72	30.86±1.66	24.73±4.08	18.77±2.70
PCA	**12.97±0.62**△	30.35±1.51△	**30.88±3.15**△	18.10±2.26△
LDA	17.56±0.45	37.60±1.10	35.80±4.01	24.35±3.70
MFA	14.34±0.62	**29.90±1.22** △	31.15±2.52	23.23±3.69
LSDA	13.27±0.48	31.79±1.42	32.33±2.70	22.26±3.74
LFDA	13.13±0.52△	30.19±1.52 △	31.15±2.98△	**17.84±2.52** △
TSD	13.29±0.90△	30.14±1.55 △	31.30±3.03△	18.55±2.74 △

two data sets probably explain why this happens. Every example in the two data sets has a sparse and binary feature vector with high dimensionality in which only a small number of elements are one, and the rest are zero. For instance, the Semeion Handwritten data set contains 1593 binary images of handwritten digits consisting 16×16 pixels. In this case, the graph Laplacian based methods may not be able to capture the meaningful local geometry of the data manifold any more. On the other hand, TSD achieves the best results probably because it can capture extra geometrical information from tangent spaces. This suggests that our proposed method makes good use of the information from tangent spaces and thus can correctly discover the data structure. In addition, the limitation inherited from the graph Laplacian based framework rather than the choice of graphs in each graph-based method should be responsible for the undesirable results, since the adopted graph in TSD is similar to those in MFA, LSDA and LFDA. Moreover, even when the feature vectors are no longer sparse and binary, TSD can also get the lowest error rates compared with the other methods with high level of statistical significance in the Image and Ionosphere data sets. This demonstrates that due to utilizing the structural information from tangent spaces, TSD can not only improve the performance of dimensionality reduction, but be applied to the data sets on which the graph Laplacian based counterparts fail to perform effectively.

Table 4 gives the time consumptions of different methods. As can be seen, TSD is relatively less efficient than its counterparts, because it has to estimate the tangent space and tangent vector at each data point. In fact, this is also the weakness of other tangent space based methods [8,13]. Therefore, proposing a strategy to make tangent space based methods more scalable can be an interesting research direction.

6 Conclusion

In this paper, we have proposed a novel supervised dimensionality reduction method named *local Tangent Space Discriminant analysis* (TSD), which differs from the methods based on the graph Laplacian framework. By introducing tangent spaces and using the first-order Taylor expansion, we develop a new strategy to utilize the information from tangent spaces, which leads to a natural way of preserving the geometrical structure of the data manifold. The proposed method aims to seek an embedding space where the local manifold structure of the data belonging to

Table 3 Mean values and standard deviations of the error rates (%) on the Image, Ionosphere, Protein Sequence, Semeion Handwritten and USPS data sets. The best method is highlighted by bold font. The best and comparable ones based on the t-test with the significance level 5% are marked with '△'.

Methods	Image	Ionosphere	Protein Sequence	Semeion	USPS
Baseline	9.25±1.00	14.00±1.98	33.17±1.05	15.98±1.16	12.07±0.72
PCA	9.09±1.00	10.77±1.98	22.43±0.83 △	13.03±1.09 △	11.62±0.64
LDA	25.99±2.31	16.94±2.86	38.97±2.99	42.94±2.15	20.26±1.47
MFA	7.57±1.03	10.66±2.16	30.21±3.20	44.95±2.26	19.46±1.23
LSDA	8.58±1.44	10.71±2.86	30.76±2.24	65.36±3.82	71.23±1.81
LFDA	7.77±0.98	13.94±1.75	40.51±6.91	45.27±2.47	12.07±1.57
TSD	**6.73±0.95**△	**9.34±1.37** △	**22.06±1.19** △	**12.78±1.04**△	**10.94±0.66**△

Table 4 Computation time (in seconds) of each method for dimensionality reduction.

Methods	PCA	LDA	MFA	LSDA	LFDA	TSD
Satellite	0.0014	0.0021	0.1198	0.0971	0.0191	5.4878
Theorem Prove	0.0015	0.0028	0.0457	0.0265	0.0098	3.8428
Breast Cancer	0.0002	0.0005	0.0039	0.0043	0.0022	0.4427
Column2C	0.0002	0.0004	0.0051	0.0043	0.0024	0.5647
Image	0.0004	0.0006	0.0232	0.0080	0.0037	1.0081
Ionosphere	0.0006	0.0010	0.0080	0.0071	0.0035	1.0006
Protein Sequence	0.1376	0.2093	0.2789	0.3219	0.2559	47.901
Semeion Handwritten	0.0400	0.0470	0.1152	0.0931	0.0914	17.103
USPS	0.0571	0.0555	0.1131	0.0974	0.1123	24.704

the same class is preserved as much as possible, while the marginal data points with different class labels are better separated. Moreover, TSD has the analytic solution by solving a generalized eigenvalue problem and can be easily extended to non-linear dimensionality reduction through the kernel trick.

The effectiveness of the proposed method has been demonstrated by comparing with related work on multiple real-world data sets including the UCI data sets and the Protein Sequence data set. The experimental results show that TSD works well on the data sets which can hardly be well handled by its counterparts, and attains better performance of classification due to utilizing the extra information from tangent spaces. Future work directions include extending our method to different learning scenarios such as semi-supervised learning and developing the sparse algorithm of TSD for large-scale learning tasks.

Acknowledgements This work is supported by the National Natural Science Foundation of China under Projects 61370175 and 61075005, and Shanghai Knowledge Service Platform Project (No.ZF1213).

References

1. K. Bache and M. Lichman. UCI machine learning repository, 2013.
2. M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.
3. D. Cai, X. He, K. Zhou, J. Han, and H. Bao. Locality sensitive discriminant analysis. In *Proceedings of the 20rd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 708–713, 2007.

4. F. R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, Rhode Island, 1997.
5. D. L. Donoho and C. Grimes. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Sciences*, 100(10):5591–5596, 2003.
6. K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, 2nd edition, 1990.
7. X. He and P. Niyogi. Locality preserving projections. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, pages 1–8. MIT Press, Cambridge, MA, 2004.
8. B. Lin, X. He, C. Zhang, and M. Ji. Parallel vector field embedding. *Journal of Machine Learning Research*, 14(1):2945–2977, 2013.
9. S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.
10. B. Schölkopf, A. Smola, and K. R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.
11. P. Simard, Y. LeCun, and J. S. Denker. Efficient pattern recognition using a new transformation distance. In S.J. Hanson, J.D. Cowan, and C.L. Giles, editors, *Advances in Neural Information Processing Systems 5*, pages 50–58. Morgan-Kaufmann, 1993.
12. M. Sugiyama. Dimensionality reduction of multimodal labeled data by local Fisher discriminant analysis. *Journal of Machine Learning Research*, 8:1027–1061, 2007.
13. S. Sun. Tangent space intrinsic manifold regularization for data representation. In *Proceedings of the IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP)*, pages 179–183, 2013.
14. H. Tyagi, E. Vural, and P. Frossard. Tangent space estimation for smooth embeddings of riemannian manifolds. *Information and Inference*, 2(1):69–114, 2013.
15. S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, and S. Lin. Graph embedding and extensions: a general framework for dimensionality reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(1):40–51, 2007.
16. J. Ye and Q. Li. A two-stage linear discriminant analysis via QR-decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):929–941, 2005.
17. Z. Zhang and H. Zha. Principal manifolds and nonlinear dimension reduction via local tangent space alignment. *SIAM Journal on Scientific Computing*, 26(1):313–338, 2004.
18. M. Zhu and A. M. Martinez. Subclass discriminant analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(8):1274–1286, 2006.

Appendix A. Detailed Derivation of S

In order to fix S, we decompose (10) into three additive terms as follows:

$$\boldsymbol{f}^\top S \boldsymbol{f} = \underbrace{\sum_{i,j=1}^{n} W_{ij}^{w} ((\boldsymbol{x}_i - \boldsymbol{x}_j)^\top \boldsymbol{t})^2}_{\text{term one}} + \underbrace{\sum_{i,j=1}^{n} W_{ij}^{w} \big(\boldsymbol{w}_{\boldsymbol{x}_j}^\top T_{\boldsymbol{x}_j}^\top (\boldsymbol{x}_i - \boldsymbol{x}_j)\big)^2}_{\text{term two}} + \underbrace{\sum_{i,j=1}^{n} W_{ij}^{w} \big[-2((\boldsymbol{x}_i - \boldsymbol{x}_j)^\top \boldsymbol{t}) \boldsymbol{w}_{\boldsymbol{x}_j}^\top T_{\boldsymbol{x}_j}^\top (\boldsymbol{x}_i - \boldsymbol{x}_j)\big]}_{\text{term three}},$$

and then examine their separate contributions to the whole S.

Term One

$$\sum_{i,j=1}^{n} W_{ij}^{w}((\boldsymbol{x}_i - \boldsymbol{x}_j)^\top \boldsymbol{t})^2$$
$$= 2\boldsymbol{t}^\top X(D^w - W^w)X^\top \boldsymbol{t} = 2\boldsymbol{t}^\top X L^w X^\top \boldsymbol{t},$$

where D^w is a diagonal weight matrix with $D_{ii}^w = \sum_{j=1}^{n} W_{ij}^w$, and $L^w = D^w - W^w$ is the Laplacian matrix. Then we have $S_1 = 2(D^w - W^w) = 2L^w$. And term one contributes to $XS_1X^\top$ in (14).

Term Two Define $B_{ji} = T_{\boldsymbol{x}_j}^\top(\boldsymbol{x}_i - \boldsymbol{x}_j)$, then

$$\sum_{i,j=1}^{n} W_{ij}^{w}\big(\boldsymbol{w}_{\boldsymbol{x}_j}^\top T_{\boldsymbol{x}_j}^\top(\boldsymbol{x}_i - \boldsymbol{x}_j)\big)^2$$
$$= \sum_{i,j=1}^{n} W_{ij}^{w}(\boldsymbol{w}_{\boldsymbol{x}_j}^\top B_{ji})^2$$
$$= \sum_{i,j=1}^{n} W_{ij}^{w}\boldsymbol{w}_{\boldsymbol{x}_j}^\top B_{ji}B_{ji}^\top \boldsymbol{w}_{\boldsymbol{x}_j}$$
$$= \sum_{j=1}^{n} \boldsymbol{w}_{\boldsymbol{x}_j}^\top \big(\sum_{i=1}^{n} W_{ij}^{w}B_{ji}B_{ji}^\top\big)\boldsymbol{w}_{\boldsymbol{x}_j} = \sum_{i=1}^{n} \boldsymbol{w}_{\boldsymbol{x}_i}^\top H_i \boldsymbol{w}_{\boldsymbol{x}_i},$$

where we have defined matrices $\{H_j\}_{j=1}^n$ with $H_j = \sum_{i=1}^{n} W_{ij}^w B_{ji}B_{ji}^\top$.

Now suppose we define a block diagonal matrix S_3 sized $mn \times mn$ with block size $m \times m$. Set the (i,i)-th block $(i = 1, \ldots, n)$ of S_3 to be H_i. Then the resultant S_3 is the contribution of term two for S in (14).

Term Three Define vectors $\{F_j\}_{j=1}^n$ with $F_j = \sum_{i=1}^{n} W_{ij}^w B_{ji}$, then term three can be rewritten as:

$$\sum_{i,j=1}^{n} W_{ij}^{w}\big[-2((\boldsymbol{x}_i - \boldsymbol{x}_j)^\top \boldsymbol{t})\boldsymbol{w}_{\boldsymbol{x}_j}^\top T_{\boldsymbol{x}_j}^\top(\boldsymbol{x}_i - \boldsymbol{x}_j)\big]$$
$$= \sum_{i,j=1}^{n} 2W_{ij}^{w}\big[((\boldsymbol{x}_j - \boldsymbol{x}_i)^\top \boldsymbol{t})\boldsymbol{w}_{\boldsymbol{x}_j}^\top B_{ji}\big]$$
$$= \sum_{i,j=1}^{n} W_{ij}^{w}(-\boldsymbol{t}^\top \boldsymbol{x}_i B_{ji}^\top \boldsymbol{w}_{\boldsymbol{x}_j}) + \sum_{i=1}^{n} \boldsymbol{t}^\top \boldsymbol{x}_i F_i^\top \boldsymbol{w}_{\boldsymbol{x}_i} +$$
$$\sum_{i,j=1}^{n} W_{ij}^{w}(-\boldsymbol{w}_{\boldsymbol{x}_j}^\top B_{ji}\boldsymbol{x}_i^\top \boldsymbol{t}) + \sum_{i=1}^{n} \boldsymbol{w}_{\boldsymbol{x}_i}^\top F_i \boldsymbol{x}_i^\top \boldsymbol{t}.$$

From this expression, we can give the formulation of S_2. Then the $S_2^\top$ in (14), which is its transpose, is ready to get.

Suppose we define two block matrices S_2^1 and S_2^2 sized $n \times mn$ each where the block size is $1 \times m$, and S_2^2 is a block diagonal matrix. Set the (i,j)-th block $(i,j = 1, \ldots, n)$ of S_2^1 to be $-W_{ij}^w B_{ji}^\top$, and the (i,i)-th block $(i = 1, \ldots, n)$ of S_2^2 to be $F_i^\top$. Then, term three can be rewritten as: $\boldsymbol{t}^\top X(S_2^1 + S_2^2)\boldsymbol{w} + \boldsymbol{w}^\top(S_2^1 + S_2^2)^\top X^\top \boldsymbol{t}$. It is clear that $S_2 = S_2^1 + S_2^2$.