Hindawi Publishing Corporation
Mathematical Problems in Engineering
Article ID 706180

Research Article

Semisupervised Tangent Space Discriminant Analysis

Yang Zhou and Shiliang Sun

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

Correspondence should be addressed to Shiliang Sun; slsun@cs.ecnu.edu.cn

Received 8 July 2014; Revised 5 November 2014; Accepted 14 November 2014

Academic Editor: Xin Xu

A novel semisupervised dimensionality reduction method named Semisupervised Tangent Space Discriminant Analysis (STSD) is presented, where we assume that data can be well characterized by a linear function on the underlying manifold. For this purpose, a new regularizer using tangent spaces is developed, which not only can capture the local manifold structure from both labeled and unlabeled data, but also has the complementarity with the Laplacian regularizer. Furthermore, STSD has an analytic form of the global optimal solution which can be computed by solving a generalized eigenvalue problem. To perform nonlinear dimensionality reduction and process structured data, a kernel extension of our method is also presented. Experimental results on multiple real-world data sets demonstrate the effectiveness of the proposed method.

1. Introduction

Dimensionality reduction is 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 representation of data will benefit subsequent tasks, for example, classification, clustering, and data visualization. Classical dimensionality reduction methods include supervised approaches like linear discriminant analysis (LDA) [1] and unsupervised ones such as principal component analysis (PCA) [2].

LDA is a supervised dimensionality reduction method. It finds a subspace in which the data points from different classes are projected far away from each other, while the data points belonging to the same class are projected as close as possible. One merit of LDA is that LDA can extract the discriminative information of data, which is crucial for classification. Due to its effectiveness, LDA is widely used in many applications, for example, bankruptcy prediction, face recognition, and data mining. However, LDA may get undesirable results when the labeled examples used for learning are not sufficient, because the between-class scatter and the within-class scatter of data could be estimated inaccurately.

PCA is a representative of unsupervised dimensionality reduction methods. It seeks a set of orthogonal projection directions along which the sum of the variances of data is maximized. PCA is a common data preprocessing technique to find a low-dimensional representation of high-dimensional data. In order to meet the requirements of different applications, many unsupervised dimensionality reduction methods have been proposed, such as Laplacian Eigenmaps [3], Hessian Eigenmaps [4], Locally Linear Embedding [5], Locality Preserving Projections [6], and Local Tangent Space Alignment [7]. Although it is shown that unsupervised approaches work well in many applications, they may not be the best choices for some learning scenarios because they may fail to capture the discriminative structure from data.

In many real-world applications, only limited labeled data can be accessed while a large number of unlabeled data are available. In this case, it is reasonable to perform semisupervised learning which can utilize both labeled and unlabeled data. Recently, several semisupervised dimensionality reduction methods have been proposed, for example, Semisupervised Discriminant Analysis (SDA) [8], Semisupervised Discriminant Analysis (SSDA) with path-based similarity [9], and Semisupervised Local Fisher Discriminant Analysis (SELF) [10]. SDA aims to find a transformation matrix following the criterion of LDA while imposing a smoothness penalty on a graph which is built to exploit the local geometry of the underlying manifold. Similarly, SSDA also builds a graph for semisupervised learning. However, the graph is constructed using a path-based similarity measure to capture the global structure of data. SELF combines the ideas of local LDA [11] and PCA so that it can integrate the information brought by both labeled and unlabeled data.

Although all of these methods have their own advantages in semisupervised learning, the essential strategy of many of them for utilizing unlabeled data relies on the Laplacian regularization. In this paper, we present a novel method named Semisupervised Tangent Space Discriminant Analysis (STSD) for semisupervised dimensionality reduction, which can reflect the discriminant information and a specific manifold structure from both labeled and unlabeled data. Unlike adopting the Laplacian based regularizer, we develop a new regularization term which can discover the linearity of the local manifold structure of data. Specifically, by introducing tangent spaces we represent the local geometry at each data point as a linear function and make the change of such functions as smooth as possible. This means that STSD appeals to a linear function on the manifold. In addition, the objective function of STSD can be optimized analytically through solving a generalized eigenvalue problem.

2. Preliminaries

Consider a data set consisting of ℓ examples and labels, $\{(\mathbf{x}_i, y_i)\}_{i=1}^{\ell}$, where $\mathbf{x}_i \in \mathbb{R}^d$ denotes a d-dimensional example, $y_i \in \{1, 2, \ldots, C\}$ denotes the class label corresponding to $\mathbf{x}_i$, and C is the total number of classes. LDA seeks a transformation $\mathbf{t}$ such that the between-class scatter is maximized and the within-class scatter is minimized [1]. The objective function of LDA can be written as

$$\mathbf{t}^{(\mathrm{LDA})} = \arg\max_{\mathbf{t}} \frac{\mathbf{t}^{\top} S_b \mathbf{t}}{\mathbf{t}^{\top} S_w \mathbf{t}}, \tag{1}$$

where $\top$ denotes the transpose of a matrix or a vector, S_b is the between-class scatter matrix, and S_w is the within-class scatter matrix. The definitions of S_b and S_w are

$$S_b = \sum_{c=1}^{C} \ell_c \left(\boldsymbol{\mu}_c - \boldsymbol{\mu}\right)\left(\boldsymbol{\mu}_c - \boldsymbol{\mu}\right)^{\top}, \tag{2}$$

$$S_w = \sum_{c=1}^{C} \sum_{\{i \mid y_i = c\}} \left(\mathbf{x}_i - \boldsymbol{\mu}_c\right)\left(\mathbf{x}_i - \boldsymbol{\mu}_c\right)^{\top}, \tag{3}$$

where ℓ_c is the number of examples from the cth class, $\boldsymbol{\mu} = (1/\ell)\sum_{i=1}^{\ell} \mathbf{x}_i$ is the mean of all the examples, and $\boldsymbol{\mu}_c = (1/\ell_c)\sum_{\{i \mid y_i = c\}} \mathbf{x}_i$ is the mean of the examples from class c.

Define the total scatter matrix as

$$S_t = \sum_{i=1}^{\ell} \left(\mathbf{x}_i - \boldsymbol{\mu}\right)\left(\mathbf{x}_i - \boldsymbol{\mu}\right)^{\top}. \tag{4}$$

It is well known that $S_t = S_b + S_w$ [1] and (1) is equivalent to

$$\mathbf{t}^{(\mathrm{LDA})} = \arg\max_{\mathbf{t}} \frac{\mathbf{t}^{\top} S_b \mathbf{t}}{\mathbf{t}^{\top} S_t \mathbf{t}}. \tag{5}$$

The solution of (5) can be readily obtained by solving a generalized eigenvalue problem: $S_b \mathbf{t} = \lambda S_t \mathbf{t}$. It should be noted that the rank of the between-class scatter matrix S_b is at most $C-1$, and thus we can obtain at most $C-1$ meaningful eigenvectors with respect to nonzero eigenvalues. This implies that LDA can project data into a space whose dimensionality is at most $C-1$.

In practice, we usually impose a regularizer on (5) to obtain a more stable solution. Then the optimization problem becomes

$$\max_{\mathbf{t}} \frac{\mathbf{t}^{\top} S_b \mathbf{t}}{\mathbf{t}^{\top} S_t \mathbf{t} + \beta R(\mathbf{t})}, \tag{6}$$

where $R(\mathbf{t})$ denotes the imposed regularizer and β is a trade-off parameter. When we use the Tikhonov regularizer, that is, $R(\mathbf{t}) = \mathbf{t}^{\top}\mathbf{t}$, the optimization problem is usually referred to as Regularized Discriminant Analysis (RDA) [12].

3. Semisupervised Tangent Space Discriminant Analysis

As a supervised method, LDA has no ability to extract information from unlabeled data. Motivated by Tangent Space Intrinsic Manifold Regularization (TSIMR) [13], we develop a novel regularizer to capture the manifold structure of both labeled and unlabeled data. Utilizing this regularizer, the LDA model can be extended to a semisupervised one following the regularization framework. Then we will first derive our novel regularizer for semisupervised learning and then present our Semisupervised Tangent Space Discriminant Analysis (STSD) algorithm as well as its kernel extension.

3.1. The Regularizer for Semisupervised Dimensionality Reduction. TSIMR [13] is a regularization method for unsupervised dimensionality reduction, which is intrinsic to data manifold and favors a linear function on the manifold. Inspired by TSIMR, we employ tangent spaces to represent the local geometry of data. Suppose that the data are sampled from an m-dimensional smooth manifold $\mathscr{M}$ in a d-dimensional space. Let $\mathscr{T}_{\mathbf{z}}\mathscr{M}$ denote the tangent space attached to $\mathbf{z}$, where $\mathbf{z} \in \mathscr{M}$ is a fixed data point on the $\mathscr{M}$. Using the first-order Taylor expansion at $\mathbf{z}$, any function f defined on the manifold $\mathscr{M}$ can be expressed as

$$f(\mathbf{x}) = f(\mathbf{z}) + \mathbf{w}_{\mathbf{z}}^{\top} \mathbf{u}_{\mathbf{z}}(\mathbf{x}) + O\left(\|\mathbf{x} - \mathbf{z}\|^2\right), \tag{7}$$

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

Consider a transformation $\mathbf{t} \in \mathbb{R}^d$ which can map the d-dimensional data to a one-dimensional embedding. Then the embedding of $\mathbf{x}$ can be expressed as $f(\mathbf{x}) = \mathbf{t}^\top \mathbf{x}$. If there are two data points $\mathbf{z}$ and $\mathbf{z}'$ that have a small Euclidean distance, by using the first-order Taylor expansion at $\mathbf{z}'$ and $\mathbf{z}$, the embeddings $f(\mathbf{z})$ and $f(\mathbf{z}')$ can be represented as

$$f(\mathbf{z}) = f(\mathbf{z}') + \mathbf{w}_{\mathbf{z}'}^\top \mathbf{u}_{\mathbf{z}'}(\mathbf{z}) + O\left(\left\|\mathbf{z} - \mathbf{z}'\right\|^2\right), \quad (8)$$

$$f(\mathbf{z}') = f(\mathbf{z}) + \mathbf{w}_{\mathbf{z}}^\top \mathbf{u}_{\mathbf{z}}(\mathbf{z}') + O\left(\left\|\mathbf{z}' - \mathbf{z}\right\|^2\right). \quad (9)$$

Suppose that the data can be well characterized by a linear function on the underlying manifold $\mathscr{M}$. Then the remainders in (8) and (9) can be omitted.

Substituting $f(\mathbf{x}) = \mathbf{t}^\top \mathbf{x}$ into (8), we have

$$\mathbf{t}^\top \mathbf{z} \approx \mathbf{t}^\top \mathbf{z}' + \mathbf{w}_{\mathbf{z}'}^\top T_{\mathbf{z}'}^\top (\mathbf{z} - \mathbf{z}'). \quad (10)$$

Furthermore, by substituting (9) into (8), we obtain

$$(T_{\mathbf{z}'} \mathbf{w}_{\mathbf{z}'} - T_{\mathbf{z}} \mathbf{w}_{\mathbf{z}})^\top (\mathbf{z} - \mathbf{z}') \approx 0, \quad (11)$$

which naturally leads to

$$T_{\mathbf{z}} \mathbf{w}_{\mathbf{z}} \approx T_{\mathbf{z}'} \mathbf{w}_{\mathbf{z}'}. \quad (12)$$

Since $T_{\mathbf{z}}$ is formed by the orthonormal bases of $\mathscr{T}_{\mathbf{z}}\mathscr{M}$, it satisfies$T_{\mathbf{z}}^\top T_{\mathbf{z}} = I_{(m\times m)}$ for all $\mathbf{z}$, where $I_{(m\times m)}$ is an m-dimensional identity matrix. We can multiply both sides of (12) with $T_{\mathbf{z}}^\top$; then (12) becomes to

$$\mathbf{w}_{\mathbf{z}} \approx T_{\mathbf{z}}^\top T_{\mathbf{z}'} \mathbf{w}_{\mathbf{z}'}. \quad (13)$$

Armed with the above results, we can formulate our regularizer for semisupervised dimensionality reduction. Consider data $\mathbf{x}_i \in X$ $(i = 1, \ldots, n)$ sampled from a function f along the manifold $\mathscr{M}$. Since every example $\mathbf{x}_i$ and its neighbors should satisfy (10) and (13), it is reasonable to formulate a regularizer as follows:

$$\begin{aligned} R(\mathbf{t}, \mathbf{w}) = \sum_{i=1}^{n} \sum_{j \in \mathscr{N}(\mathbf{x}_i)} \Big[\Big(\mathbf{t}^\top (\mathbf{x}_i - \mathbf{x}_j) \\ - \mathbf{w}_{\mathbf{x}_j}^\top T_{\mathbf{x}_j}^\top (\mathbf{x}_i - \mathbf{x}_j) \Big)^2 \\ + \gamma \left\| \mathbf{w}_{\mathbf{x}_i} - T_{\mathbf{x}_i}^\top T_{\mathbf{x}_j} \mathbf{w}_{\mathbf{x}_j} \right\|_2^2 \Big], \end{aligned} \quad (14)$$

where $\mathbf{w} = (\mathbf{w}_{\mathbf{x}_1}^\top, \mathbf{w}_{\mathbf{x}_2}^\top, \ldots, \mathbf{w}_{\mathbf{x}_n}^\top)^\top$, $\mathscr{N}(\mathbf{x}_i)$ denotes the set of nearest neighbors of $\mathbf{x}_i$, and γ is a trade-off parameter to control the influences of (10) and (13).

Relating data with a discrete weighted graph is a popular choice, and there are indeed a large family of graph based statistical and machine learning methods. It also makes sense for us to generalize the regularizer $R(\mathbf{t}, \mathbf{w})$ in (14) using a symmetric weight matrix W constructed from the above data collection X. There are several manners to construct W. One typical way is to build an adjacency graph by connecting each data point to its k-nearest-neighbors with an edge and then weight every edge of the graph by a certain measure. Generally, if two data points $\mathbf{x}_i$ and $\mathbf{x}_j$ are "close," the corresponding weight W_{ij} is large, whereas if they are "far away," then the W_{ij} is small. For example, the heat kernel function is widely used to construct a weight matrix. The weight W_{ij} is computed by

$$W_{ij} = \exp\left(- \frac{\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2}{\sigma^2} \right), \quad (15)$$

if there is an edge connecting $\mathbf{x}_i$ with $\mathbf{x}_j$ and $W_{ij} = 0$ otherwise.

Therefore, the generalization of the proposed regularizer turns out to be

$$\begin{aligned} R(\mathbf{t}, \mathbf{w}) = \sum_{i=1}^{n} \sum_{j=1}^{n} W_{ij} \Big[\Big(\mathbf{t}^\top (\mathbf{x}_i - \mathbf{x}_j) \\ - \mathbf{w}_{\mathbf{x}_j}^\top T_{\mathbf{x}_j}^\top (\mathbf{x}_i - \mathbf{x}_j) \Big)^2 \\ + \gamma \left\| \mathbf{w}_{\mathbf{x}_i} - T_{\mathbf{x}_i}^\top T_{\mathbf{x}_j} \mathbf{w}_{\mathbf{x}_j} \right\|_2^2 \Big], \end{aligned} \quad (16)$$

and W is an $n \times n$ symmetric weight matrix reflecting the similarity of the data points. It is clear that when the variation of the first-order Taylor expansion at every data point is smooth, the value of $R(\mathbf{t}, \mathbf{w})$, which measures the linearity of the function f along the manifold $\mathscr{M}$, will be small.

The regularizer (16) can be reformulated as a canonical matrix quadratic form as follows:

$$\begin{aligned} R(\mathbf{t}, \mathbf{w}) &= \begin{pmatrix} \mathbf{t} \\ \mathbf{w} \end{pmatrix}^\top S \begin{pmatrix} \mathbf{t} \\ \mathbf{w} \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{t} \\ \mathbf{w} \end{pmatrix}^\top \begin{pmatrix} X S_1 X^\top & X S_2 \\ S_2^\top X^\top & S_3 \end{pmatrix} \begin{pmatrix} \mathbf{t} \\ \mathbf{w} \end{pmatrix}, \end{aligned} \quad (17)$$

where $X = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ is the data matrix and S is a positive semidefinite matrix constructed by four blocks, that is, $XS_1X^\top$, XS_2, $S_2^\top X^\top$, and S_3. This formulation will be very useful in developing our algorithm. Recall that the dimensionality of the directional derivative $\mathbf{w}_{\mathbf{x}_i}$ $(i = 1, \ldots, n)$ is m. Thereby the size of S is $(d+mn)\times(d+mn)$. For simplicity, we omit the detailed derivation of S.

It should be noted that, besides the principle that accorded with TSIMR, the regularizer (16) can be explained from another perspective. Recently, Lin et al. [14] proposed a regularization method called Parallel Field Regularization (PFR) for semisupervised regression. In spite of the different learning scenarios, PFR shares the same spirit with TSIMR in essence. Moreover, when the bases of the tangent space $\mathscr{T}_{\mathbf{z}}\mathscr{M}$ at any data point $\mathbf{z}$ are orthonormal, PFR can be converted to TSIMR. It also provides a more theoretical but complex explanation for our regularizer from the vector field perspective.

Input: Labeled and unlabeled examples $\{(\mathbf{x}_i, y_i) \mid \mathbf{x}_i \in \mathbb{R}^d, y_i \in \{1, 2, \dots, C\}\}_{i=1}^{\ell}$, $\{\mathbf{x}_i \mid \mathbf{x}_i \in \mathbb{R}^d\}_{i=\ell+1}^{n}$;
Trade-off parameters α, β, γ ($\alpha, \beta, \gamma \geq 0$).
Output: $d \times C$ transformation matrix T.
Construct the adjacency graph;
Calculate the weight matrix W;
for $i = 1$ **to** n **do**
  Construct $T_{\mathbf{x}_i}$ using local PCA;
end for
Compute the eigenvectors $\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_C$ of (20) with respect to the non-zero eigenvalues;
$T = (\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_C)$.

Algorithm 1: STSD.

3.2. An Algorithm. With the regularizer developed in Section 3.1, we can present our STSD algorithm. Suppose the training data include ℓ labeled examples $\{(\mathbf{x}_i, y_i)\}_{i=1}^{\ell}$ belonging to C classes and $n-\ell$ unlabeled examples $\{\mathbf{x}_i\}_{i=\ell+1}^{n}$ where $\mathbf{x}_i \in \mathbb{R}^d$ is a d-dimensional example and $y_i \in \{1, 2, \dots, C\}$ is the class label associated with the example $\mathbf{x}_i$. Define $\mathbf{f} = (\mathbf{t}^\top, \mathbf{w}^\top)^\top$, and let $\widetilde{S}_b = \begin{pmatrix} S_b & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}$, $\widetilde{S}_t = \begin{pmatrix} S_t & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}$ be two $(d+mn) \times (d+mn)$ augmented matrices extended from the between-class scatter matrix S_b and the total scatter matrix S_t. Note that in the semisupervised learning scenario discussed in this section, the mean of all the samples in (2) and (4) should be the center of both the labeled and unlabeled examples; that is, $\boldsymbol{\mu} = (1/n)\sum_{i=1}^{n} \mathbf{x}_i$. The objective function of STSD can be written as follows:

$$\max_{\mathbf{f}} \frac{\mathbf{f}^\top \widetilde{S}_b \mathbf{f}}{\mathbf{f}^\top \left(\widetilde{S}_t + \alpha S\right) \mathbf{f}}, \tag{18}$$

where α is a trade-off parameter. It is clear that $\mathbf{f}^\top \widetilde{S}_b \mathbf{f} = \mathbf{t}^\top S_b \mathbf{t}$ and $\mathbf{f}^\top \widetilde{S}_t \mathbf{f} = \mathbf{t}^\top S_t \mathbf{t}$. Therefore, STSD seeks an optimal $\mathbf{f}$ such that the between-class scatter is maximized, and the total scatter as well as the regularizer $R(\mathbf{t}, \mathbf{w})$ defined in (17) is minimized at the same time.

The optimization of the objective function (18) can be achieved by solving a generalized eigenvalue problem:

$$\widetilde{S}_b \mathbf{f} = \lambda \left(\widetilde{S}_t + \alpha S\right) \mathbf{f} \tag{19}$$

whose solution can be easily given by the eigenvector with respect to the maximal eigenvalue. Note that since the mean $\mathbf{u}$ is the center of both labeled and unlabeled examples, the rank of $\widetilde{S}_b$ is C. It implies that there are at most C eigenvectors with respect to the nonzero eigenvalues. Therefore, given the optimal eigenvectors $\mathbf{f}_1, \dots, \mathbf{f}_C$, we can form a transformation matrix sized $d \times C$ as $T = (\mathbf{t}_1, \dots, \mathbf{t}_C)$, and then the C-dimensional embedding $\mathbf{b}$ of an example $\mathbf{x}$ can be computed through $\mathbf{b} = T^\top \mathbf{x}$.

In many applications, especially when the dimensionality of data is high while the data size is small, the matrix $\widetilde{S}_t + \alpha S$ in (19) may be singular. This singularity problem may lead to an unstable solution and deteriorate the performance of STSD. Fortunately, there are many approaches to deal with the singularity problem. In this paper, we use the Tikhonov regularization because of its simplicity and wide applicability. Finally, the generalized eigenvalue problem (19) turns out to be

$$\widetilde{S}_b \mathbf{f} = \lambda \left(\widetilde{S}_t + \alpha S + \beta I\right) \mathbf{f}, \tag{20}$$

where I is the identity matrix and $\beta \geq 0$. Algorithm 1 gives the pseudocode for STSD.

The main computational cost of STSD lies in building tangent spaces for n data points and solving the generalized eigenvalue problem (20). The naive implementation for our algorithm has a runtime of $O((d^2 m + m^2 d) \times n)$ for the construction of tangent spaces and $O((d+mn)^3)$ for the generalized eigenvalue decomposition. This suggests that STSD might be a time-consuming method.

However, given a neighborhood size k, there are only $k+1$ examples as the inputs of local PCA. Then we can obtain at most $k+1$ meaningful orthonormal bases to construct each tangent space, which implies that the dimensionality m of the directional derivative $\mathbf{w}_{\mathbf{x}_i}$ ($i = 1, \dots, n$) is always less than $k+1$. In practice, k is usually small to ensure the locality. This makes sure that m is actually a small constant. Furthermore, recall that the number of eigenvectors with respect to nonzero eigenvalues is equal to the number of classes C. Using the technique of sparse generalized eigenvalue decomposition, the corresponding computational cost is reduced to $O(C^2 \times (d+mn))$.

In summary, the overall runtime of STSD is $O((d^2 m + m^2 d) \times n + C^2 \times (d+mn))$. Since m and C are always small, STSD actually has an acceptable computational cost.

3.3. Kernel STSD. Essentially STSD is a linear dimensionality reduction method, which can not be used for nonlinear dimensionality reduction or processing structured data such as graphs, trees, or other types of structured inputs. To handle this problem, we extend STSD to a Reproducing Kernel Hilbert Space (RKHS).

Suppose examples $\mathbf{x}_i \in \mathscr{X}$ $(i = 1, \ldots, n)$, where $\mathscr{X}$ is an input domain. Consider a feature space $\mathscr{F}$ induced by a nonlinear mapping $\phi : \mathscr{X} \to \mathscr{F}$. We can construct an RKHS $H_{\mathscr{K}}$ by defining a kernel function $\mathscr{K}(\cdot,\cdot)$ using the inner product operation $\langle\cdot,\cdot\rangle$, such that $\mathscr{K}(\mathbf{x},\mathbf{y}) = \langle\phi(\mathbf{x}),\phi(\mathbf{y})\rangle$. Let $\Phi_l = (\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_\ell))$, $\Phi_u = (\phi(\mathbf{x}_{\ell+1}), \ldots, \phi(\mathbf{x}_n))$ be the labeled and unlabeled data matrix in the feature space $\mathscr{F}$, respectively. Then the total data matrix can be written as $\Phi = (\Phi_l, \Phi_u)$.

Let $\phi(\boldsymbol{\mu})$ be the mean of all the examples in $\mathscr{F}$ and define $\Psi = (\phi(\boldsymbol{\mu}_1), \ldots, \phi(\boldsymbol{\mu}_C))$ which is constituted by the mean vectors of each class in $\mathscr{F}$. Suppose that $\phi(\boldsymbol{\mu}) = 0$ (it can be easily achieved by centering the data in the feature space) and the labeled examples in Φ_l are ordered according to their labels. Then the between-class scatter matrix S_b^ϕ and the total scatter matrix S_t^ϕ in $\mathscr{F}$ can be written as $S_b^\phi = \Psi M \Psi^\top$, $S_t^\phi = \Phi \widetilde{I} \Phi^\top$ where M is a $C \times C$ diagonal matrix whose (c,c)th element is the number of the examples belonging to class c and $\widetilde{I} = \begin{pmatrix} I_{\ell\times\ell} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}$ is a $n \times n$ matrix where $I_{\ell\times\ell}$ is the identity matrix sized $\ell \times \ell$.

Recall that STSD aims to find a set of transformations to map data into a low-dimensional space. Given examples $\mathbf{x}_1, \ldots, \mathbf{x}_n$, one can use the orthogonal projection to decompose any transformation $\mathbf{t} \in H_{\mathscr{K}}$ into a sum of two functions: one lying in the *span*$\{\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_n)\}$ and the other one lying in the orthogonal complementary space. Therefore, there exist a set of coefficients α_i $(i = 1, 2, \ldots, n)$ satisfying

$$\mathbf{t} = \sum_{i=1}^{n} \alpha_i \phi(\mathbf{x}_i) + \mathbf{v} = \Phi\boldsymbol{\alpha} + \mathbf{v}, \tag{21}$$

where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_n)^\top$ and $\langle\mathbf{v}, \phi(\mathbf{x}_i)\rangle = 0$ for all i. Note that although we set $\mathbf{f} = (\mathbf{t}^\top, \mathbf{w}^\top)^\top$ and optimize $\mathbf{t}$ and $\mathbf{w}$ together, there is no need to reparametrize $\mathbf{w}$ like $\mathbf{t}$. What we need is to estimate tangent spaces in $\mathscr{F}$ through local Kernel PCA [15].

Let $T_{\mathbf{x}_i}^\phi$ be the matrix formed by the orthonormal bases of the tangent space attached to $\phi(\mathbf{x}_i)$. Substitute (21) into (17) and replace $T_{\mathbf{x}_i}$ with $T_{\mathbf{x}_i}^\phi$ $(i = 1, 2, \ldots, n)$. We can reformulate the regularizer (17) as follows:

$$\begin{aligned} R(\boldsymbol{\alpha}, \mathbf{w}) &= \boldsymbol{\alpha}^\top \Phi^\top \Phi S_1 \Phi^\top \Phi \boldsymbol{\alpha} + \mathbf{w}^\top S_3 \mathbf{w} \\ &\quad + \boldsymbol{\alpha}^\top \Phi^\top \Phi S_2 \mathbf{w} + \mathbf{w}^\top S_2^\top \Phi^\top \Phi \boldsymbol{\alpha} \\ &= \boldsymbol{\alpha}^\top K S_1 K \boldsymbol{\alpha} + \mathbf{w}^\top S_3 \mathbf{w} \\ &\quad + \boldsymbol{\alpha}^\top K S_2 \mathbf{w} + \mathbf{w}^\top S_2^\top K \boldsymbol{\alpha}, \end{aligned} \tag{22}$$

where K is a kernel matrix with $K_{ij} = \mathscr{K}(\mathbf{x}_i, \mathbf{x}_j)$. With this formulation, Kernel STSD can be converted to a generalized eigenvalue problem as follows:

$$\widetilde{S}_b^\phi \boldsymbol{\varphi} = \lambda \left(\widetilde{S}_t^\phi + \alpha S^\phi\right) \boldsymbol{\varphi}, \tag{23}$$

where we have defined $\boldsymbol{\varphi} = (\boldsymbol{\alpha}^\top, \mathbf{w}^\top)^\top$. The definitions of $\widetilde{S}_b^\phi$, $\widetilde{S}_t^\phi$, and S^ϕ are given as follows:

$$\begin{aligned} \widetilde{S}_b^\phi &= \begin{pmatrix} \Phi^\top S_b^\phi \Phi & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} = \begin{pmatrix} \Phi^\top \Psi M \Psi^\top \Phi & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}, \\ \widetilde{S}_t^\phi &= \begin{pmatrix} \Phi^\top S_t^\phi \Phi & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} = \begin{pmatrix} K \widetilde{I} K & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}, \\ S^\phi &= \begin{pmatrix} K S_1 K & K S_2 \\ S_2^\top K & S_3 \end{pmatrix}. \end{aligned} \tag{24}$$

It should be noted that every term of $\mathbf{v}$ vanishes from the formulation of Kernel STSD because $\langle\mathbf{v}, \phi(\mathbf{x}_i)\rangle = 0$ for all i. Since $\Psi^\top\Phi$ can be computed through the kernel matrix K, the solution of Kernel STSD can be obtained without knowing the explicit form of the mapping ϕ.

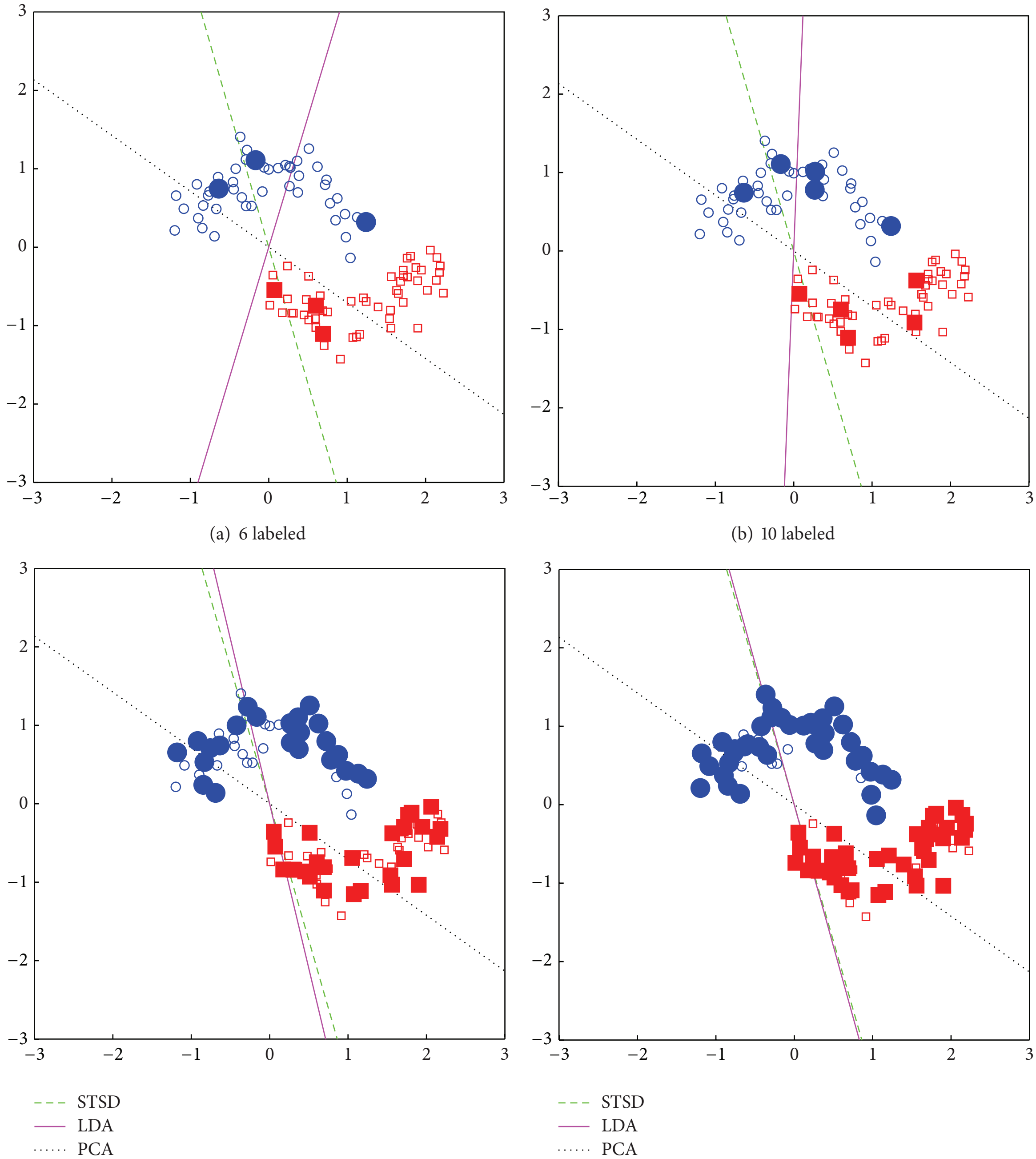
Given the eigenvectors $\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_C$ with respect to the nonzero eigenvalues of (23), the resulting transformation matrix can be written as $\Gamma = (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_C)$. Then, the embedding $\mathbf{b}$ of an original example $\mathbf{x}$ can be computed as

$$\mathbf{b} = \Gamma^\top \Phi^\top \phi(\mathbf{x}) = \Gamma^\top \left(\mathscr{K}(\mathbf{x}_1, \mathbf{x}), \ldots, \mathscr{K}(\mathbf{x}_n, \mathbf{x})\right)^\top. \tag{25}$$

4. Experiments

4.1. Toy Data. In order to illustrate the behavior of STSD, we first perform STSD on a toy data set (two moons) compared with PCA and LDA. The toy data set contains 100 data points and is used under different label configurations. Specifically, 6, 10, 50, and 80 data points are randomly labeled, respectively, and the rest are unlabeled, where PCA is trained by all the data points without labels, LDA is trained by labeled data only, and STSD is trained by both the labeled and unlabeled data. In Figure 1, we show the one-dimensional embedding spaces found by different methods (onto which data points will be projected). As can be seen in Figure 1(a), although LDA is able to find an optimum projection where the within-class scatter is minimized while the between-class separability is maximized, it can hardly find a good projection when the labeled data are scarce. In addition, PCA also finds a bad solution, since it has no ability to utilize the discriminant information from class labels. On the contrary, STSD, which can utilize both the labeled and unlabeled data, finds a desirable projection onto which data from different classes have the minimal overlap. As the number of labeled data increases, we can find that the solutions of PCA and STSD do not change, while the projections found by PCA are gradually close to those of STSD. In Figure 1(d), the solutions of LDA and STSD are almost identical, which means that by utilizing both labeled and unlabeled data, STSD can obtain the optimum solutions even when only a few data points are labeled. This demonstrates the usefulness and advantage of STSD in the semisupervised scenario.

4.2. Real-World Data. In this section, we evaluate STSD with real-world data sets. Specifically, we first perform dimensionality reduction to map all examples into a subspace and then carry out classification using the nearest neighbor classifier



(a) 6 labeled

(b) 10 labeled

(c) 50 labeled

(d) 80 labeled

Figure 1: Illustrative examples of STSD, LDA, and PCA on the two-moon data set under different label configurations. The circles and squares denote the data points in positive and negative classes, and the filled or unfilled symbols denote the labeled or unlabeled data, respectively.

(1-NN) in the subspace. This measurement for evaluating semisupervised dimensionality reduction methods is widely used in literature, such as [8–10, 16]. For each data set, we randomly split out 80% of the data as the training set and the rest as the test set. In the training set, a certain number of data are randomly labeled while the rest of the data are unlabeled. Moreover, every experimental result is obtained from the average over 20 splits.

In our experiments, we compare STSD with multiple dimensionality reduction methods including PCA, LDA, SELF, and SDA, where LDA is performed only on the labeled data, while PCA, SELF, SDA, and STSD are performed on both the labeled and unlabeled data. In addition, we also compare our method with the baseline method which just employs the 1-NN classifier with the labeled data in the original space. Since the performances of PCA and SELF

Table 1: Mean values and standard deviations of the unlabeled error rates (%) with different label configurations on the face data sets.

Method	Yale 3	Yale 4	ORL 2	ORL 3
Baseline	49.50 ± 4.86	43.93 ± 4.71	30.31 ± 3.11	21.13 ± 2.29
PCA	47.67 ± 4.40	42.60 ± 5.05	29.23 ± 2.56	20.30 ± 2.22
LDA	32.56 ± 3.85	25.60 ± 2.98	17.17 ± 3.23	8.05 ± 2.51
SELF	54.22 ± 3.88	52.07 ± 4.67	48.79 ± 4.39	37.48 ± 2.81
SDA	32.33 ± 4.11	25.93 ± 3.22	16.67 ± 3.36	7.85 ± 2.48
STSD	**32.28 ± 4.09**	**25.27 ± 3.61**	**16.00 ± 3.03**	**7.73 ± 2.30**

Table 2: Mean values and standard deviations of the test error rates (%) with different label configurations on the face data sets.

Method	Yale 3	Yale 4	ORL 2	ORL 3
Baseline	46.17 ± 7.67	46.67 ± 8.65	29.94 ± 3.66	19.19 ± 3.50
PCA	40.67 ± 8.06	42.00 ± 7.29	28.06 ± 3.92	18.13 ± 3.71
LDA	32.33 ± 8.31	26.17 ± 7.74	16.56 ± 3.97	9.13 ± 3.63
SELF	50.00 ± 6.49	49.33 ± 8.28	47.88 ± 4.82	35.56 ± 3.52
SDA	32.00 ± 8.40	26.17 ± 7.67	16.13 ± 4.05	9.00 ± 3.33
STSD	**31.83 ± 8.41**	**25.33 ± 8.54**	**15.69 ± 3.68**	**9.00 ± 3.16**

depend on the dimensionality of the embedding subspace discovered by each method, we show the best results for them.

For the graph based methods, including SELF, SDA, and STSD, the number of nearest neighbors for constructing adjacency graphs is determined by fourfold cross-validation. The parameters α and γ for STSD are selected through fourfold cross-validation, while the Tikhonov regularization parameter β is fixed to 10^{-1}. In addition, the parameters involved in SELF and SDA are also selected through fourfold cross-validation. We use the heat kernel function (15) to construct the weight matrix, and the kernel parameter σ^2 is fixed as d_{av} unless otherwise specified where d_{av} is the average of the squared distances between all data points and their nearest neighbors.

Two types of data sets under different label configurations are used to conduct our experiments. One type of data sets is the face images which consist of high-dimensional images, and the other one is the UCI data sets constituted by low-dimensional data. For the convenience of description, we name each configuration of experiments as "Data Set" + "Labeled Data Size." For example, for the experiments with the face images, "Yale 3" means the experiment is performed on the Yale data set with 3 labeled data per class. Analogously, for the experiments with the UCI data sets, "BCWD 20" means the experiment is performed on the Breast Cancer Wisconsin (Diagnostic) data set with a total of 20 labeled examples from all classes.

4.2.1. Face Images. It is well known that high-dimensional data such as images and texts are supposed to live on or near a low-dimensional manifold. In this section, we test our algorithm with the Yale and ORL face data sets which are deemed to satisfy this manifold assumption. The Yale data set contains 165 images of 15 individuals and there are 11 images per subject. The images have different facial expressions, illuminations, and facial details (with or without glass). The ORL data set contains 400 images of 40 distinct subjects under varying expressions and illuminations. In our experiments, every face image is cropped to consist of 32×32 pixels with 256 grey levels per pixel. Furthermore, for the Yale data set, we set the parameter σ^2 of the heat kernel to $0.1 d_{\mathrm{av}}$. We report the error rates on both the unlabeled training data and test data. Tables 1 and 2 show that STSD is always better than or at least comparable with other counterparts in all the cases, which demonstrates that STSD can well exploit the manifold structure for dimensionality reduction. Notice that SELF gets inferior results. We conjecture that this is because it has no ability to capture the underlying manifold structures of the data.

4.2.2. UCI Data Sets. In this set of experiments, we use three UCI data sets [17] including Breast Cancer Wisconsin (Diagnostic), Climate Model Simulation Crashes, and Cardiotocography which may not well satisfy the manifold assumption. For simplicity, we abbreviate these data sets as BCWD, CMSC, and CTG, respectively. BCWD consists of 569 data points from two classes in $\mathbb{R}^{30}$. CMSC consists of 540 data points from two classes in $\mathbb{R}^{18}$. CTG consists of 2126 data points from ten classes in $\mathbb{R}^{23}$.

From the results reported in Tables 3 and 4, it can be seen that when the labeled data are scarce, the performance of LDA is even worse than the baseline method due to the inaccurate estimation of the scatter matrices. However, STSD achieves the best or comparable results among all other methods in all configurations, expect for the test error rate in BCWD 10. Although STSD adopts a relatively strong manifold assumption, it still has sufficient flexibility to handle general data which may not live on a low-dimensional manifold.

Table 3: Mean values and standard deviations of the unlabeled error rates (%) with different label configurations on the UCI data sets.

Method	BCWD 10	BCWD 30	CMSC 10	CMSC 30	CTG 20	CTG 160
Baseline	11.90 ± 4.04	10.22 ± 3.21	14.39 ± 6.40	14.05 ± 1.90	63.71 ± 3.73	47.91 ± 1.73
PCA	11.87 ± 4.01	10.21 ± 3.27	11.86 ± 2.51	13.43 ± 2.40	63.74 ± 3.75	47.89 ± 1.66
LDA	20.34 ± 8.76	9.61 ± 2.76	13.18 ± 4.49	14.21 ± 3.28	67.28 ± 6.32	41.60 ± 2.65
SELF	13.43 ± 3.63	14.1 ± 4.20	10.06 ± 3.30	11.88 ± 2.53	67.00 ± 4.50	44.09 ± 2.66
SDA	10.10 ± 3.26	7.12 ± 2.17	9.06 ± 0.97	8.71 ± 0.78	58.27 ± 5.01	41.91 ± 2.17
STSD	**10.07 ± 3.46**	**6.99 ± 1.73**	**8.98 ± 1.04**	**8.60 ± 0.58**	**58.11 ± 4.78**	**40.88 ± 2.15**

Table 4: Mean values and standard deviations of the test error rates (%) with different label configurations on the UCI data sets.

Method	BCWD 10	BCWD 30	CMSC 10	CMSC 30	CTG 20	CTG 160
Baseline	12.75 ± 6.56	10.65 ± 4.20	14.63 ± 8.48	12.81 ± 4.37	64.15 ± 4.74	48.76 ± 2.44
PCA	12.75 ± 6.56	10.50 ± 4.16	8.75 ± 1.90	9.13 ± 2.66	64.07 ± 4.76	48.66 ± 2.41
LDA	20.60 ± 10.34	10.85 ± 5.06	13.19 ± 5.66	15.13 ± 5.35	67.47 ± 7.27	41.95 ± 3.43
SELF	14.65 ± 6.88	13.70 ± 4.37	9.06 ± 2.66	8.69 ± 1.84	67.02 ± 5.06	43.34 ± 2.81
SDA	**9.75 ± 3.60**	8.75 ± 3.09	8.69 ± 3.15	8.06 ± 1.70	58.72 ± 4.26	41.67 ± 3.11
STSD	10.15 ± 4.55	**8.50 ± 3.19**	**8.63 ± 2.59**	**8.06 ± 1.54**	**58.62 ± 4.11**	**41.55 ± 3.31**

Notice that the error rates of several dimensionality reduction methods over the CMSC data set do not improve with the increasing size of labeled data. The reason may be that the data in the CMSC data set contain some irrelevant features as reflected by the original data description [18], which leads to the unexpected results. Nevertheless, SDA and STSD achieve more reasonable results due to their capabilities to extract information from both labeled and unlabeled data.

It should be noted that overall the experiments are conducted with 5 data sets, and in terms of the results of all the data sets STSD is likely to beat other methods account for a sign-test's P value of 0.031, which is statistically significant. This also demonstrates that STSD is better than the related methods.

4.3. Connection with the Laplacian Regularization. Essentially, both STSD and SDA are regularized LDA methods with specific regularizers. STSD imposes the regularizer (16) which prefers a linear function along the manifold, while SDA employs the Laplacian regularizer to penalize the function differences among "similar" examples. Now consider a regularized LDA method using both of these regularizers named STSLap, whose objective function can be written as follows:

$$\max_{\mathbf{t},\mathbf{w}} \frac{\mathbf{t}^{\top} S_b \mathbf{t}}{\mathbf{t}^{\top} S_t \mathbf{t} + \overline{\alpha} R_{\mathrm{Lap}}(\mathbf{t}) + \overline{\beta} R_{\mathrm{STS}}(\mathbf{t},\mathbf{w})}, \quad (26)$$

where $R_{\mathrm{Lap}}(\mathbf{t}) = \mathbf{t}^{\top} L \mathbf{t}$ is the Laplacian regularizer used in SDA with L being the Laplacian matrix [19] and $R_{\mathrm{STS}}(\mathbf{t},\mathbf{w})$ is the regularizer used in STSD, which is defined as (16). The parameters $\overline{\alpha}$ and $\overline{\beta}$ are used to control the trade-off between the influences of $R_{\mathrm{Lap}}(\mathbf{t})$ and $R_{\mathrm{STS}}(\mathbf{t},\mathbf{w})$. Similar to STSD, STSLap can also be converted to a generalized eigenvalue problem, which can be easily solved through eigenvalue decomposition.

Although the previous experiments have shown that STSD gets better results than SDA in most situations, SDA can achieve similar results with STSD in some configurations. However, this does not mean that STSD and SDA are similar or, in other words, $R_{\mathrm{STS}}(\mathbf{t},\mathbf{w})$ and $R_{\mathrm{Lap}}(\mathbf{t})$ have similar behavior. In fact, the two regularizers seem to complement each other. To demonstrate this complementarity, we compare STSLap with SDA and STSD under a medium-sized label configuration over all the data sets used in the previous experiments. Specifically, the experiments are performed on BCWD 30, CMSC 30, CTG 160, Yale 3, and ORL 2. For each data set, the neighborhood size used to construct the adjacency graph is set to be the one supported by the experimental results with both SDA and STSD in Sections 4.2.1 and 4.2.2. This means that all the methods compared in this section utilize the same graph to regularize the LDA model for each data set. The parameters $\overline{\alpha}$, $\overline{\beta}$ in (26) and γ in $R_{\mathrm{STS}}(\mathbf{t},\mathbf{w})$ are selected through fourfold cross-validation.

Note that given a graph, the performance of STSLap can be at least, ideally, identical to SDA or STSD, because STSLap degenerates to SDA or STSD when the parameter $\overline{\alpha}$ or $\overline{\beta}$ is set to zero. However, if STSLap achieves better results than both SDA and STSD, we can deem that $R_{\mathrm{Lap}}(\mathbf{t})$ and $R_{\mathrm{STS}}(\mathbf{t},\mathbf{w})$ are complementary.

Tables 5 and 6 show that the performance of STSLap is better than both SDA and STSD in most of the cases. Moreover, although it is not shown in the tables, the trade-off parameters $\overline{\alpha}$ and $\overline{\beta}$ are scarcely set to be zero by cross-validation. This means that STSLap always utilizes the information discovered from both $R_{\mathrm{Lap}}(\mathbf{t})$ and $R_{\mathrm{STS}}(\mathbf{t},\mathbf{w})$. In conclusion, the proposed regularizer $R_{\mathrm{STS}}(\mathbf{t},\mathbf{w})$ can capture the manifold structure of data which can not be discovered by Laplacian regularizer. This implies that these two regularizers are complementary to each other, and we could use them together to yield probably better results in practice. It should

Table 5: Mean values and standard deviations of the unlabeled error rates (%) with medium-sized labeled data on different data sets.

Method	BCWD 30	CMSC 30	CTG 160	Yale 3	ORL 2
SDA	**6.88 ± 2.53**	9.60 ± 2.27	41.97 ± 2.72	**32.39 ± 5.98**	20.81 ± 2.76
STSD	6.96 ± 2.45	**9.40 ± 2.30**	43.47 ± 2.83	32.56 ± 6.67	16.48 ± 2.14
STSLap	7.07 ± 2.46	9.60 ± 2.24	**41.57 ± 2.66**	33.39 ± 7.01	**16.42 ± 2.07**

Table 6: Mean values and standard deviations of the test error rates (%) with medium-sized labeled data on different data sets.

Method	BCWD 30	CMSC 30	CTG 160	Yale 3	ORL 2
SDA	6.90 ± 2.86	9.56 ± 3.28	41.85 ± 3.23	33.33 ± 5.92	20.63 ± 5.98
STSD	6.70 ± 2.81	9.44 ± 3.45	42.47 ± 3.57	33.00 ± 6.20	14.81 ± 4.20
STSLap	**6.45 ± 2.74**	**9.38 ± 3.13**	**41.18 ± 3.54**	**32.83 ± 6.24**	**14.44 ± 4.28**

be noted that our aim is not to compare STSD with SDA in this set of experiments, and we can not make any conclusion about whether or not STSD is better than SDA from Tables 5 and 6 because the neighbourhood size for each data set is fixed.

5. Discussion

5.1. Related Work. STSD is a semisupervised dimensionality reduction method under a certain manifold assumption. More specifically, we assume that the distribution of data can be well approximated by a linear function on the underlying manifold. One related method named SDA [8] adopts another manifold assumption. It simply assumes that the mapping function should be as smooth as possible on a given graph. This strategy is well known as the Laplacian regularization which is widely employed in the semisupervised learning scenario. However, STSD follows a different principle to regularize the mapping function, which not only provides an alternative strategy for semisupervised dimensionality reduction, but also attains the complementarity with the classic Laplacian regularization. SELF [10] is another related approach, which is a hybrid method of local LDA [11] and PCA. Despite its simplicity, SELF can only discover the linear structure of data, whereas our method is able to capture the nonlinear intrinsic manifold structure.

Rather than constructing an appropriate regularizer on a given graph, SSDA [9] and semisupervised dimensionality reduction (SSDR) [16] focus on building a good graph and then perform the Laplacian-style regularization on this graph. SSDA regularizes LDA on a graph constructed by a path-based similarity measure. The advantage of SSDA is its robustness against outliers, because SSDA aims to preserve the global manifold information. SSDR constructs a graph according to the so-called must-link and cannot-link pairwise constraints, which gives a natural way to incorporate prior knowledge into the semisupervised dimensionality reduction. However, this prior knowledge is not always available in practice. In contrast to SSDA and SSDR, our method is flexible enough to perform regularization on any graph and free from the necessity of extra prior knowledge. In fact, the advantage of SSDA or SSDR can be easily inherited through performing STSD with the graph constructed by corresponding method (SSDA or SSDR), which is another important merit of STSD.

5.2. Further Improvements. For the manifold related learning problem considered in STSD, the estimation of bases for tangent spaces is an important step. In this paper, we use local PCA with fixed neighborhood size to calculate the tangent spaces, and the neighborhood size is set to be same as the one used to construct the adjacency graph. This is certainly not the optimal choice, since manifolds can have varying curvatures and data could be nonuniformly sampled. Note that the neighborhood size can determine the evolution of calculated tangent spaces along the manifold. When a small neighborhood size k is used, there are at most $k+1$ examples for the inputs of local PCA. However, when we need to estimate a set of tangent spaces which have relative high dimensionality m ($m > k + 1$), it is almost impossible to get accurate estimates of the tangent spaces, because there are at most $k+1$ meaningful orthonormal bases obtained from local PCA. Moreover, noises can damage the manifold assumption as well to a certain extent. All these factors explain the necessity for using different neighborhood sizes and more robust subspace estimation methods.

In our method, each example in the data matrix can be treated as an anchor point, where local PCA is used to calculate the tangent space. The number of parameters that should be estimated in our method basically grows linearly with respect to the number of anchor points. Therefore, in order to reduce the parameters to be estimated, one possible approach is to reduce the anchor points where only “key” examples are kept as the anchor points. This will be a kind of research for data set sparsification. People can make different criteria to decide whether or not an example should be regarded as the “key” one.

The research of anchor point reduction is especially useful when training data are large-scale. For large-scale data, anchor point reduction can be promising to speed up the training process. In addition, data can exhibit different manifold dimensions at different regions, especially for complex data. Therefore, adaptively determining the dimensionality at different anchor points is also an important refinement of the current approach.

6. Conclusion

In this paper, we have proposed a novel semisupervised dimensionality reduction method named Semisupervised Tangent Space Discriminant Analysis (STSD), which can extract the discriminant information as well as the manifold structure from both labeled and unlabeled data, where a linear function assumption on the manifold is exploited. Local PCA is involved as an important step to estimate tangent spaces and certain relationships between adjacent tangent spaces are derived to reflect the adopted model assumption. The optimization of STSD is readily achieved by the eigenvalue decomposition.

Experimental results on multiple real-world data sets including the comparisons with related works have shown the effectiveness of the proposed method. Furthermore, the complementarity between our method and the Laplacian regularization has also been verified. Future work directions include finding more accurate methods for tangent space estimation and extending our method to different learning scenarios such as multiview learning and transfer learning.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

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

References

[1] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, 2nd edition, 1990.

[2] I. T. Jolliffe, *Principal Component Analysis*, Springer, New York, NY, USA, 1986.

[3] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[4] D. L. Donoho and C. Grimes, "Hessian eigenmaps: locally linear embedding techniques for high-dimensional data," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 100, no. 10, pp. 5591–5596, 2003.

[5] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[6] X. He and P. Niyogi, "Locality preserving projections," in *Advances in Neural Information Processing Systems*, S. Thrun, L. Saul, and B. Scholkopf, Eds., vol. 18, pp. 1–8, MIT Press, Cambridge, Mass, USA, 2004.

[7] Z. Zhang and H. Zha, "Principal manifolds and nonlinear dimensionality reduction via tangent space alignment," *SIAM Journal on Scientific Computing*, vol. 26, no. 1, pp. 313–338, 2004.

[8] D. Cai, X. He, and J. Han, "Semi-supervised discriminant analysis," in *Proceedings of the 11th IEEE International Conference on Computer Vision (ICCV '07)*, pp. 1–7, Rio de Janeiro, Brazil, October 2007.

[9] Y. Zhang and D.-Y. Yeung, "Semi-supervised discriminant analysis using robust path-based similarity," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1–8, Anchorage, Alaska, USA, June 2008.

[10] M. Sugiyama, T. Ide, S. Nakajima, and J. Sese, "Semi-supervised local Fisher discriminant analysis for dimensionality reduction," *Machine Learning*, vol. 78, no. 1-2, pp. 35–61, 2010.

[11] M. Sugiyama, "Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis," *The Journal of Machine Learning Research*, vol. 8, pp. 1027–1061, 2007.

[12] J. H. Friedman, "Regularized discriminant analysis," *Journal of the American Statistical Association*, vol. 84, no. 405, pp. 165–175, 1989.

[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 '13)*, pp. 179–183, Beijing, China, July 2013.

[14] B. Lin, C. Zhang, and X. He, "Semi-supervised regression via parallel field regularization," in *Advances in Neural Information Processing Systems*, J. Shawe-Taylor, R. S. Zemel, P. Bartlett, F. C. N. Pereira, and K. Q. Weinberger, Eds., vol. 24, pp. 433–441, The MIT Press, Cambridge, Mass, USA, 2011.

[15] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, 1998.

[16] D. Zhang, Z. Zhou, and S. Chen, "Semi-supervised dimensionality reduction," in *Proceedings of the 7th SIAM International Conference on Data Mining*, pp. 629–634, April 2007.

[17] K. Bache and M. Lichman, UCI machine learning repository, 2013, http://archive.ics.uci.edu/ml/.

[18] D. D. Lucas, R. Klein, J. Tannahill et al., "Failure analysis of parameter-induced simulation crashes in climate models," *Geoscientific Model Development Discussions*, vol. 6, no. 1, pp. 585–623, 2013.

[19] F. R. K. Chung, *Spectral Graph Theory*, American Mathematical Society, Providence, RI, USA, 1997.