Semi-Supervised Support Vector Machines with Tangent Space Intrinsic Manifold Regularization

Shiliang Sun and Xijiong Xie

Abstract—Semi-supervised learning has been an active research topic in machine learning and data mining. One main reason is that labeling examples is expensive and time-consuming while there are large numbers of unlabeled examples available in many practical problems. So far, Laplacian regularization has been widely used in semi-supervised learning. In this paper, we propose a new regularization method called tangent space intrinsic manifold regularization. It is intrinsic to data manifold and favors linear functions on the manifold. Fundamental elements involved in the formulation of the regularization are local tangent space representations which are estimated by local principal component analysis, and the connections which relate adjacent tangent spaces. Simultaneously, we explore its application to semi-supervised classification and propose two new learning algorithms called tangent space intrinsic manifold regularized support vector machines (TiSVMs) and tangent space intrinsic manifold regularized twin support vector machines (TiTSVMs). They effectively integrate the tangent space intrinsic manifold regularization consideration. The optimization of TiSVMs can be solved by a standard quadratic programming while the optimization of TiTSVMs can be solved by a pair of standard quadratic programmings. Experimental results on semi-supervised classification problems show the effectiveness of the proposed semi-supervised learning algorithms.

Index Terms—Support vector machine, twin support vector machine, semi-supervised classification, manifold learning, tangent space intrinsic manifold regularization

1 Introduction

Semi-supervised classification, which estimates a decision function from few labeled examples and a large quantity of unlabeled examples, is an active research topic. Its prevalence is mainly motivated by the need to reduce the expensive or time-consuming label acquisition process. Evidence shows that, provided that the unlabeled data which are inexpensive to collect are properly exploited, people can obtain a superior performance over the counterpart supervised learning approaches with few labeled examples. For a comprehensive survey of semi-supervised learning methods, refer to [1] and [2].

Current semi-supervised classification methods can be divided into two categories, which are called single-view and multi-view algorithms, respectively. Their difference lies in the number of feature sets used to train classifiers. If more than one feature set is adopted to learn classifiers, the algorithm would be called a multi-view semi-supervised learning algorithm. The Laplacian support vector machines (LapSVMs) [3], [4] and Laplacian twin support vector machines (LapTSVMs) [5], which can be regarded as two applications of Laplacian eigenmaps to semi-supervised learning, are representative algorithms for single-view semi-supervised classification. Typical multi-view classification methods include co-training [6], SVM-2K [7], co-Laplacian SVMs [8], manifold co-regularization [9], multi-view Laplacian SVMs [10], sparse multi-view SVMs [11] and multi-view Laplacian TSVMs [12]. Although the regularization method presented in this paper can be applied to multi-view semi-supervised classification after some appropriate manipulation, in this paper we only focus on the single-view classification problem.

The principle of regularization has its root in mathematics to solve ill-posed problems [13], and is widely used in statistics and machine learning [3], [14], [15]. Many well-known algorithms, e.g., SVMs [16], TSVMs [17], ridge regression and lasso [18], can be interpreted as instantiations of the idea of regularization. A close parallel to regularization is the capacity control of function classes [19]. Both regularization and the capacity control can alleviate the ill-posed and overfitting problems of learning algorithms. Moreover, from the point of view of Bayesian learning, the solution to a regularization problem corresponds to the maximum a posterior (MAP) estimate for a parameter of interest. The regularization term plays the role of the prior distribution on the parameter in the Bayesian model [20].

In many real applications, data lying in a high-dimensional space can be assumed to be intrinsically of low dimensionality. That is, data can be well characterized by far fewer parameters or degrees of freedom than the actual ambient representation. This setting is usually referred to as manifold learning, and the distribution of data is regarded to live on or near a low-dimensional manifold. The validity of manifold learning

Manuscript received 6 Dec. 2014; revised 7 May and 18 Jul. 2015; accepted 23 Jul. 2015. This work was supported by the National Natural Science Foundation of China under Project 61370175 and the Science and Technology Commission of Shanghai Municipality under Grant 14DZ2260800.

Shiliang Sun and Xijiong Xie are with the 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. E-mail: slsun@cs.ecnu.edu.cn.

has already been testified by some recent developments, e.g., the work in [21], [22] and [23]. Laplacian regularization is an important manifold learning method, which is used to exploit the geometry of the probability distribution by assuming that its support has the geometric structure of a Riemannian manifold. Graph-based learning methods often use this regularization to obtain an approximation to the underlying manifold. In particular, Laplacian regularization has been widely used in semi-supervised learning to effectively combine labeled examples and unlabeled examples. For examples, LapSVMs and LapTSVMs are two representative semi-supervised classification methods with Laplacian regularization. Laplacian regularized least squares are regarded as a representative semi-supervised regression method with Laplacian regularization [3]. The optimization of such algorithms is built on a representer theorem that provides a basis for many algorithms for unsupervised, semi-supervised and fully supervised learning.

In this paper, we propose a new regularization method called tangent space intrinsic manifold regularization to approximate a manifold more subtly. Through this regularization we can learn a linear function $f(x)$ on the manifold. The new regularization has potentials to be applied to a variety of statistical and machine learning problems. In later descriptions, Laplacian regularization is in fact only a part of the tangent space intrinsic manifold regularization.

Part of this research has been reported in a short conference paper [24]. Compared to the previous work, we have derived the formulation of the new regularization in detail. While the previous work mainly considered data representation with the new regularization, this paper considers a different task semi-supervised classification and exhibits the usefulness of the new regularization method for this task. Two new learning machines TiSVMs and TiTSVMs are thus proposed. TiSVMs integrate the common hinge loss for classification, norm regularization, and the tangent space intrinsic manifold regularization term, and lead to a quadratic programming problem, while TiTSVMs lead to a pair of quadratic programming problems. Semi-supervised classification experiments with TiSVMs and TiTSVMs on multiple datasets give encouraging results.

The remainder of this paper is organized as follows. In Section 2, we introduce the methodology of the tangent space intrinsic manifold regularization. Then in Section 3 we generalize it based upon the popular weighted-graph representation of manifolds from data, and reformulate the regularization term as a matrix quadratic form through which we can gain insights about the regularization and draw connections with related regularization methods. Moreover, this reformulation will facilitate resolving semi-supervised classification tasks, and the corresponding algorithms are given in Section 4 and Section 5, respectively. Experimental results which shows the effectiveness of the regularization method in semi-supervised classification are reported in Section 6, followed by Section 7 which discusses further refinements about the proposed methods and other possible applications. Concluding remarks are given in Section 8.

2 Methodology of the Tangent Space Intrinsic Manifold Regularization

We are interested in estimating a function $f(\mathbf{x})$ defined on $\mathcal{M} \subset \mathbb{R}^d$, where $\mathcal{M}$ is a smooth manifold on $\mathbb{R}^d$. We assume that $f(\mathbf{x})$ can be well approximated by a linear function with respect to the manifold $\mathcal{M}$. Let m be the dimensionality of $\mathcal{M}$. At each point $\mathbf{z} \in \mathcal{M}$, $f(\mathbf{x})$ can be represented as a linear function $f(\mathbf{x}) \approx b_{\mathbf{z}} + \mathbf{w}_{\mathbf{z}}^\top \mathbf{u}_{\mathbf{z}}(\mathbf{x}) + o(\|\mathbf{x}-\mathbf{z}\|^2)$ locally around $\mathbf{z}$, where $\mathbf{u}_{\mathbf{z}}(\mathbf{x}) = T_{\mathbf{z}}(\mathbf{x}-\mathbf{z})$ is an m-dimensional vector representing $\mathbf{x}$ in the tangent space around $\mathbf{z}$, and $T_{\mathbf{z}}$ is an $m \times d$ matrix that projects $\mathbf{x}$ around $\mathbf{z}$ to a representation in the tangent space of $\mathcal{M}$ at $\mathbf{z}$. Note that in this paper the basis for $T_{\mathbf{z}}$ is computed using local principal component analysis (PCA) for its simplicity and wide applicability. In particular, the point $\mathbf{z}$ and its neighbors are sent over to the regular PCA procedure [25] and the top m eigenvectors of the $d \times d$ covariance matrix are returned back as rows of matrix $T_{\mathbf{z}}$. The weight vector $\mathbf{w}_{\mathbf{z}} \in \mathbb{R}^m$ is an m-dimensional vector, and it is also the manifold-derivative of $f(\mathbf{x})$ at $\mathbf{z}$ with respect to the $\mathbf{u}_{\mathbf{z}}(\cdot)$ representation on the manifold, which we write as $\nabla_T f(\mathbf{x})|_{\mathbf{x}=\mathbf{z}} = \mathbf{w}_{\mathbf{z}}$.

Mathematically, a linear function with respect to the manifold $\mathcal{M}$, which is not necessarily a globally linear function in $\mathbb{R}^d$, is a function that has constant manifold derivative. However, this does not mean $\mathbf{w}_{\mathbf{z}}$ is a constant function of $\mathbf{u}$ due to the different coordinate systems when the "anchor point" $\mathbf{z}$ changes from one point to another. This needs to be compensated using "connections" that map a coordinate representation $\mathbf{u}_{\mathbf{z}'}$ to $\mathbf{u}_{\mathbf{z}}$ for any $\mathbf{z}'$ near $\mathbf{z}$. For points far apart, the connections are not of interest for our purpose, since coordinate systems on a manifold usually change and representing distant points with a single basis would thereby lead to a large bias.

To see how our approach works, we assume for simplicity that $T_{\mathbf{z}}$ is an orthogonal matrix for all $\mathbf{z}$: $T_{\mathbf{z}} T_{\mathbf{z}}^\top = I_{(m\times m)}$. This means that if $\mathbf{x} \in \mathcal{M}$ is close to $\mathbf{z} \in \mathcal{M}$, then $\mathbf{x}-\mathbf{z} \approx T_{\mathbf{z}}^\top T_{\mathbf{z}}(\mathbf{x}-\mathbf{z}) + O(\|\mathbf{x}-\mathbf{z}\|^2)$. Now consider $\mathbf{x}$ that is close to both $\mathbf{z}$ and $\mathbf{z}'$. We can express $f(\mathbf{x})$ both in the tangent space representation at $\mathbf{z}$ and $\mathbf{z}'$, which gives

$$b_{\mathbf{z}} + \mathbf{w}_{\mathbf{z}}^\top \mathbf{u}_{\mathbf{z}}(\mathbf{x}) \approx b_{\mathbf{z}'} + \mathbf{w}_{\mathbf{z}'}^\top \mathbf{u}_{\mathbf{z}'}(\mathbf{x}) + O(\|\mathbf{x}-\mathbf{z}'\|^2 + \|\mathbf{x}-\mathbf{z}\|^2).$$

That is, $b_{\mathbf{z}} + \mathbf{w}_{\mathbf{z}}^\top \mathbf{u}_{\mathbf{z}}(\mathbf{x}) \approx b_{\mathbf{z}'} + \mathbf{w}_{\mathbf{z}'}^\top \mathbf{u}_{\mathbf{z}'}(\mathbf{x})$. This means that

$$b_{\mathbf{z}} + \mathbf{w}_{\mathbf{z}}^\top T_{\mathbf{z}}(\mathbf{x}-\mathbf{z}) \approx b_{\mathbf{z}'} + \mathbf{w}_{\mathbf{z}'}^\top T_{\mathbf{z}'}(\mathbf{x}-\mathbf{z}').$$

Setting $\mathbf{x}=\mathbf{z}$, we obtain $b_{\mathbf{z}} \approx b_{\mathbf{z}'} + \mathbf{w}_{\mathbf{z}'}^\top T_{\mathbf{z}'}(\mathbf{z}-\mathbf{z}')$, and

$$b_{\mathbf{z}'} + \mathbf{w}_{\mathbf{z}'}^\top T_{\mathbf{z}'}(\mathbf{z}-\mathbf{z}') + \mathbf{w}_{\mathbf{z}}^\top T_{\mathbf{z}}(\mathbf{x}-\mathbf{z}) \approx b_{\mathbf{z}'} + \mathbf{w}_{\mathbf{z}'}^\top T_{\mathbf{z}'}(\mathbf{x}-\mathbf{z}').$$

This implies that

$$\mathbf{w}_{\mathbf{z}}^\top T_{\mathbf{z}}(\mathbf{x}-\mathbf{z}) \approx \mathbf{w}_{\mathbf{z}'}^\top T_{\mathbf{z}'}(\mathbf{x}-\mathbf{z}) \approx \mathbf{w}_{\mathbf{z}'}^\top T_{\mathbf{z}'} T_{\mathbf{z}}^\top T_{\mathbf{z}}(\mathbf{x}-\mathbf{z}) + O(\|\mathbf{x}-\mathbf{z}'\|^2 + \|\mathbf{x}-\mathbf{z}\|^2). \quad (1)$$

Since (1) holds for arbitrary $\mathbf{x} \in \mathcal{M}$ close to $\mathbf{z} \in \mathcal{M}$, it follows that, $\mathbf{w}_{\mathbf{z}}^\top \approx \mathbf{w}_{\mathbf{z}'}^\top T_{\mathbf{z}'} T_{\mathbf{z}}^\top + O(\|\mathbf{z}-\mathbf{z}'\|)$ or $\mathbf{w}_{\mathbf{z}} \approx T_{\mathbf{z}} T_{\mathbf{z}'}^\top \mathbf{w}_{\mathbf{z}'} + O(\|\mathbf{z}-\mathbf{z}'\|)$.

This means that if we expand at points $\mathbf{z}_1, \ldots, \mathbf{z}_k \in Z$, and denote neighbors of $\mathbf{z}_j$ as $\mathcal{N}(\mathbf{z}_j)$, then the correct regularizer will be

$$R(\{b_{\mathbf{z}}, \mathbf{w}_{\mathbf{z}}\}_{\mathbf{z}\in Z}) = \sum_{i=1}^{k} \sum_{j\in\mathcal{N}(\mathbf{z}_i)} \Big[\big(b_{\mathbf{z}_i} - b_{\mathbf{z}_j} - \mathbf{w}_{\mathbf{z}_j}^\top T_{\mathbf{z}_j} (\mathbf{z}_i - \mathbf{z}_j) \big)^2 + \gamma \|\mathbf{w}_{\mathbf{z}_i} - T_{\mathbf{z}_i} T_{\mathbf{z}_j}^\top \mathbf{w}_{\mathbf{z}_j}\|_2^2 \Big] . \tag{2}$$

The function $f(\mathbf{x})$ is approximated as follows

$$f(\mathbf{x}) = b_{\mathbf{z}(\mathbf{x})} + \mathbf{w}_{\mathbf{z}(\mathbf{x})}^\top T_{\mathbf{z}(\mathbf{x})} \big(\mathbf{x} - \mathbf{z}(\mathbf{x}) \big), \tag{3}$$

where $\mathbf{z}(\mathbf{x}) = \arg\min_{\mathbf{z}\in Z} \|\mathbf{x} - \mathbf{z}\|_2$. This is a very natural formulation for out-of-example extensions.

2.1 Effect of Local PCA

As we consider the setting of manifold learning and the dimensionality of the tangent space is usually less than the one of the outer space, the local PCA is used to determine the local tangent space. If we don't use local PCA and consider the original space, the above approach is equivalent to considering a piecewise linear function, namely $T_{\mathbf{z}} = T_{\mathbf{z}}^\top = I$. The corresponding expression of the regularization becomes

$$R(\{b_{\mathbf{z}}, \mathbf{w}_{\mathbf{z}}\}_{\mathbf{z}\in Z}) = \sum_{i=1}^{k} \sum_{j\in\mathcal{N}(\mathbf{z}_i)} \Big[\big(b_{\mathbf{z}_i} - b_{\mathbf{z}_j} - \mathbf{w}_{\mathbf{z}_j}^\top (\mathbf{z}_i - \mathbf{z}_j) \big)^2 + \gamma \|\mathbf{w}_{\mathbf{z}_i} - \mathbf{w}_{\mathbf{z}_j}\|_2^2 \Big] . \tag{4}$$

However, this will lead to higher dimensionality and more parameters.

The rationality of the proposed regularization principle can be interpreted from the standpoint of effective function learning. A globally linear function in the original ambient Euclidean space is often too simple, but instead a locally or piecewise linear function in this space would be too complex, since it can have too many parameters to be estimated. The linear function with respect to the manifold as favored by the current regularization method is a good trade-off between these two situations. Basically it can be seen as a locally linear function in the ambient space, but since the dimensionality of the function weights is m rather than d, the number of parameters to be learned is greatly reduced compared to the local linear function setting in the original Euclidean space. This reflects a good leverage between flexibility and manageability for effective function learning from data.

3 Generalization and Reformulation of the Regularization Term

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 $R(\{b_{\mathbf{z}}, \mathbf{w}_{\mathbf{z}}\}_{\mathbf{z}\in Z})$ in (2) using a symmetric weight matrix W constructed from the above data collection Z.

Entries in W characterize the closeness of different points where the points are often called nodes in the terminology of graphs. Usually there are two steps involved in constructing a weighted graph. The first step builds an adjacency graph by putting an edge between two "close" points. People can choose to use parameter $\epsilon \in \mathbb{R}$ or parameter $n \in \mathbb{N}$ to determine close points, which means that two nodes would be connected if their Euclidean distance is within ϵ or either node is among the n nearest neighbors of the other as indicated by the Euclidean distance. The second step calculates weights on the edges of the graph with a certain similarity measure. For example, the heat-kernel method computes weight W_{ij} for two connected nodes i and j by

$$W_{ij} = \exp^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{t}} \tag{5}$$

where parameter $t > 0$, while for nodes not directly connected the weights would be zero [23]. One can also use the polynomial kernel to calculate weights $W_{ij} = (\mathbf{x}_i^\top \mathbf{x}_j)^p$ where parameter $p \in \mathbb{N}$ is the polynomial degree. The simplest approach for weight assignment is to adopt the $\{0, 1\}$ values where weights are 1 for connected edges and 0 for others.

Therefore, the generalization of the tangent space intrinsic manifold regularizer turns out to be

$$R(\{b_{\mathbf{z}}, \mathbf{w}_{\mathbf{z}}\}_{\mathbf{z}\in Z}) = \sum_{i=1}^{k} \sum_{j=1}^{k} W_{ij} \Big[\big(b_{\mathbf{z}_i} - b_{\mathbf{z}_j} - \mathbf{w}_{\mathbf{z}_j}^\top T_{\mathbf{z}_j} (\mathbf{z}_i - \mathbf{z}_j) \big)^2 + \gamma \|\mathbf{w}_{\mathbf{z}_i} - T_{\mathbf{z}_i} T_{\mathbf{z}_j}^\top \mathbf{w}_{\mathbf{z}_j}\|_2^2 \Big] . \tag{6}$$

The previous regularizer is a special case of (6) with $\{0, 1\}$ weights. The advantage of the new regularizer is that the discrepancy of the neighborhood relationship is treated more logically.

Now we reformulate the regularizer (6) into a canonical matrix quadratic form. The benefits include relating our regularization method with other regularization approaches and facilitating subsequent formulations on semi-supervised classification. In particular, we would like to rewrite the regularizer as a quadratic form in terms of a symmetric matrix S as follows,

$$R(\{b_{\mathbf{z}}, \mathbf{w}_{\mathbf{z}}\}_{\mathbf{z}\in Z}) = \begin{pmatrix} b_{\mathbf{z}_1} \\ \vdots \\ b_{\mathbf{z}_k} \\ \mathbf{w}_{\mathbf{z}_1} \\ \vdots \\ \mathbf{w}_{\mathbf{z}_k} \end{pmatrix}^\top S \begin{pmatrix} b_{\mathbf{z}_1} \\ \vdots \\ b_{\mathbf{z}_k} \\ \mathbf{w}_{\mathbf{z}_1} \\ \vdots \\ \mathbf{w}_{\mathbf{z}_k} \end{pmatrix}$$

$$= \begin{pmatrix} b_{\mathbf{z}_1} \\ \vdots \\ b_{\mathbf{z}_k} \\ \mathbf{w}_{\mathbf{z}_1} \\ \vdots \\ \mathbf{w}_{\mathbf{z}_k} \end{pmatrix}^\top \begin{pmatrix} S_1 & S_2 \\ S_2^\top & S_3 \end{pmatrix} \begin{pmatrix} b_{\mathbf{z}_1} \\ \vdots \\ b_{\mathbf{z}_k} \\ \mathbf{w}_{\mathbf{z}_1} \\ \vdots \\ \mathbf{w}_{\mathbf{z}_k} \end{pmatrix}, \tag{7}$$

where $\begin{pmatrix} S_1 & S_2 \\ S_2^\top & S_3 \end{pmatrix}$ is a block matrix representation of matrix S and the size of S_1 is $k \times k$. In this paper, we suppose that $\mathbf{w}_{\mathbf{z}_i}$ $(i = 1, \ldots, k)$ is an m-dimensional vector. Therefore, the size of S is $(k+mk) \times (k+mk)$. In order to fix S, we decompose (6) into four additive terms as follows and then examine their separate contributions to the whole S

$$
\begin{aligned}
R(\{b_{\mathbf{z}}, \mathbf{w}_{\mathbf{z}}\}_{\mathbf{z}\in Z}) = & \underbrace{\sum_{i=1}^{k}\sum_{j=1}^{k} W_{ij}(b_{\mathbf{z}_i} - b_{\mathbf{z}_j})^2}_{\text{term one}} \\
& + \underbrace{\sum_{i=1}^{k}\sum_{j=1}^{k} W_{ij}\big(\mathbf{w}_{\mathbf{z}_j}^\top T_{\mathbf{z}_j}(\mathbf{z}_i - \mathbf{z}_j)\big)^2}_{\text{term two}} \\
& + \underbrace{\sum_{i=1}^{k}\sum_{j=1}^{k} W_{ij}\big[-2(b_{\mathbf{z}_i} - b_{\mathbf{z}_j})\mathbf{w}_{\mathbf{z}_j}^\top T_{\mathbf{z}_j}(\mathbf{z}_i - \mathbf{z}_j)\big]}_{\text{term three}} \\
& + \underbrace{\gamma\sum_{i=1}^{k}\sum_{j=1}^{k} W_{ij}\|\mathbf{w}_{\mathbf{z}_i} - T_{\mathbf{z}_i}T_{\mathbf{z}_j}^\top \mathbf{w}_{\mathbf{z}_j}\|_2^2}_{\text{term four}} \ .
\end{aligned}
$$

3.1 Term One

$$
\sum_{i=1}^{k}\sum_{j=1}^{k} W_{ij}(b_{\mathbf{z}_i} - b_{\mathbf{z}_j})^2 = 2\begin{pmatrix} b_{\mathbf{z}_1} \\ \vdots \\ b_{\mathbf{z}_k} \end{pmatrix}^{\top}(D - W)\begin{pmatrix} b_{\mathbf{z}_1} \\ \vdots \\ b_{\mathbf{z}_k} \end{pmatrix},
$$

where D is a diagonal weight matrix with $D_{ii} = \sum_{j=1}^{k} W_{ij}$. This means that term one contributes to S_1 in (7). Actually, we have $S_1 = 2(D - W)$.

3.2 Term Two

Define vector $B_{ji} = T_{\mathbf{z}_j}(\mathbf{z}_i - \mathbf{z}_j)$. Then,

$$
\begin{aligned}
& \sum_{i=1}^{k}\sum_{j=1}^{k} W_{ij}\big(\mathbf{w}_{\mathbf{z}_j}^\top T_{\mathbf{z}_j}(\mathbf{z}_i - \mathbf{z}_j)\big)^2 \\
= & \sum_{i=1}^{k}\sum_{j=1}^{k} W_{ij}(\mathbf{w}_{\mathbf{z}_j}^\top B_{ji})^2 \\
= & \sum_{i=1}^{k}\sum_{j=1}^{k} W_{ij}\mathbf{w}_{\mathbf{z}_j}^\top B_{ji}B_{ji}^\top \mathbf{w}_{\mathbf{z}_j} \\
= & \sum_{j=1}^{k} \mathbf{w}_{\mathbf{z}_j}^\top\big(\sum_{i=1}^{k} W_{ij}B_{ji}B_{ji}^\top\big)\mathbf{w}_{\mathbf{z}_j} = \sum_{i=1}^{k} \mathbf{w}_{\mathbf{z}_i}^\top H_i \mathbf{w}_{\mathbf{z}_i},
\end{aligned}
$$

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

Now suppose we define a block diagonal matrix S_3^H sized $mk \times mk$ with block size $m \times m$. Set the (i,i)-th block $(i = 1, \ldots, k)$ of S_3^H to be H_i. Then the resultant S_3^H is the contribution of term two for S_3 in (7).

3.3 Term Three

$$
\begin{aligned}
& \sum_{i=1}^{k}\sum_{j=1}^{k} W_{ij}\big[-2(b_{\mathbf{z}_i} - b_{\mathbf{z}_j})\mathbf{w}_{\mathbf{z}_j}^\top T_{\mathbf{z}_j}(\mathbf{z}_i - \mathbf{z}_j)\big] \\
= & \sum_{i=1}^{k}\sum_{j=1}^{k} W_{ij}\big[-2(b_{\mathbf{z}_i} - b_{\mathbf{z}_j})\mathbf{w}_{\mathbf{z}_j}^\top B_{ji}\big] \\
= & \ 2\sum_{i=1}^{k}\sum_{j=1}^{k} W_{ij}(-b_{\mathbf{z}_i}\mathbf{w}_{\mathbf{z}_j}^\top B_{ji}) + 2\sum_{i=1}^{k}\sum_{j=1}^{k} W_{ij}(b_{\mathbf{z}_j}\mathbf{w}_{\mathbf{z}_j}^\top B_{ji}) \\
= & \sum_{i=1}^{k}\sum_{j=1}^{k} W_{ij}(-b_{\mathbf{z}_i}B_{ji}^\top \mathbf{w}_{\mathbf{z}_j}) + \sum_{i=1}^{k}\sum_{j=1}^{k} W_{ij}(-\mathbf{w}_{\mathbf{z}_j}^\top B_{ji}b_{\mathbf{z}_i}) \\
& + \sum_{j=1}^{k} b_{\mathbf{z}_j}\big(\sum_{i=1}^{k} W_{ij}B_{ji}^\top\big)\mathbf{w}_{\mathbf{z}_j} + \sum_{j=1}^{k} \mathbf{w}_{\mathbf{z}_j}^\top\big(\sum_{i=1}^{k} W_{ij}B_{ji}\big)b_{\mathbf{z}_j} \\
= & \sum_{i=1}^{k}\sum_{j=1}^{k} W_{ij}(-b_{\mathbf{z}_i}B_{ji}^\top \mathbf{w}_{\mathbf{z}_j}) + \sum_{i=1}^{k} b_{\mathbf{z}_i}F_i^\top \mathbf{w}_{\mathbf{z}_i} \\
& + \sum_{i=1}^{k}\sum_{j=1}^{k} W_{ij}(-\mathbf{w}_{\mathbf{z}_j}^\top B_{ji}b_{\mathbf{z}_i}) + \sum_{i=1}^{k} \mathbf{w}_{\mathbf{z}_i}^\top F_i b_{\mathbf{z}_i},
\end{aligned}
$$

where we have defined vectors $\{F_j\}_{j=1}^{k}$ with $F_j = \sum_{i=1}^{k} W_{ij}B_{ji}$. From this equation, we can give the formulation of S_2, and then the $S_2^\top$ in (7), which is its transpose, is ready to get.

Suppose we define two block matrices S_2^1 and S_2^2 sized $k \times mk$ each where the block size is $1 \times m$, and S_2^2 is a block diagonal matrix. Set the (i,j)-th block $(i,j = 1, \ldots, k)$ of S_2^1 to be $-W_{ij}B_{ji}^\top$, and the (i,i)-th block $(i = 1, \ldots, k)$ of S_2^2 to be $F_i^\top$. Then, it is clear that $S_2 = S_2^1 + S_2^2$.

3.4 Term Four

Denote matrix $T_{z_i}T_{z_j}^\top$ by A_{ij}. Then, with γ omitted temporarily,

$$
\begin{aligned}
& \sum_{i=1}^{k}\sum_{j=1}^{k} W_{ij}\|\mathbf{w}_{\mathbf{z}_i} - T_{\mathbf{z}_i}T_{\mathbf{z}_j}^\top \mathbf{w}_{\mathbf{z}_j}\|_2^2 \\
= & \sum_{i=1}^{k}\sum_{j=1}^{k} W_{ij}\|\mathbf{w}_{\mathbf{z}_i} - A_{ij}\mathbf{w}_{\mathbf{z}_j}\|_2^2 \\
= & \sum_{i=1}^{k}\sum_{j=1}^{k} W_{ij}\mathbf{w}_{\mathbf{z}_i}^\top \mathbf{w}_{\mathbf{z}_i} + \sum_{i=1}^{k}\sum_{j=1}^{k} W_{ij}\mathbf{w}_{\mathbf{z}_j}^\top A_{ij}^\top A_{ij}\mathbf{w}_{\mathbf{z}_j} \\
& - \sum_{i=1}^{k}\sum_{j=1}^{k} 2W_{ij}\mathbf{w}_{\mathbf{z}_i}^\top A_{ij}\mathbf{w}_{\mathbf{z}_j} \\
= & \sum_{i=1}^{k} D_{ii}\mathbf{w}_{\mathbf{z}_i}^\top I_{(m\times m)}\mathbf{w}_{\mathbf{z}_i} + \sum_{j=1}^{k} \mathbf{w}_{\mathbf{z}_j}^\top\big(\sum_{i=1}^{k} W_{ij}A_{ij}^\top A_{ij}\big)\mathbf{w}_{\mathbf{z}_j} \\
& - \sum_{i=1}^{k}\sum_{j=1}^{k} 2W_{ij}\mathbf{w}_{\mathbf{z}_i}^\top A_{ij}\mathbf{w}_{\mathbf{z}_j} \\
= & \sum_{i=1}^{k} \mathbf{w}_{\mathbf{z}_i}^\top(D_{ii}I + C_i)\mathbf{w}_{\mathbf{z}_i} - \sum_{i=1}^{k}\sum_{j=1}^{k} \mathbf{w}_{\mathbf{z}_i}^\top(2W_{ij}A_{ij})\mathbf{w}_{\mathbf{z}_j},
\end{aligned}
$$

where in the last line we have defined matrices $\{C_j\}_{j=1}^k$ with $C_j = \sum_{i=1}^k W_{ij} A_{ij}^\top A_{ij}$.

Now suppose we define two block matrices S_3^1 and S_3^2 sized $mk \times mk$ each where the block size is $m \times m$, and S_3^1 is a block diagonal matrix. Set the (i,i)-th block $(i = 1, \ldots, k)$ of S_3^1 to be $D_{ii} I + C_i$, and the (i,j)-th block $(i,j = 1, \ldots, k)$ of S_3^2 to be $2W_{ij} A_{ij}$. Then the contribution of term four for S_3 would be $\gamma(S_3^1 - S_3^2)$. Further considering the contribution of term two for S_3, we finally have $S_3 = S_3^H + \gamma(S_3^1 - S_3^2)$.

3.5 Connections with the Laplacian Regularization

From our reformulation, we can draw a connection between our regularizer and the Laplacian regularization [23]. The Laplacian regularizer, in our current terminology, can be expressed as

$$\begin{aligned} & \sum_{i=1}^k \sum_{j=1}^k W_{ij} (b_{\mathbf{z}_i} - b_{\mathbf{z}_j})^2 \\ = \;& 2 \begin{pmatrix} b_{\mathbf{z}_1} \\ \vdots \\ b_{\mathbf{z}_k} \end{pmatrix}^\top (D - W) \begin{pmatrix} b_{\mathbf{z}_1} \\ \vdots \\ b_{\mathbf{z}_k} \end{pmatrix} \\ = \;& 2 \begin{pmatrix} b_{\mathbf{z}_1} \\ \vdots \\ b_{\mathbf{z}_k} \end{pmatrix}^\top L \begin{pmatrix} b_{\mathbf{z}_1} \\ \vdots \\ b_{\mathbf{z}_k} \end{pmatrix}, \end{aligned}$$

where $L = D - W$ is the Laplacian matrix. Obviously, the matrix S_1 in the tangent space intrinsic manifold regularizer equals $2L$. In this sense, we can say that our regularizer reflects the Laplacian regularization to a certain extent. However, this regularizer is more complicated as it intends to favor linear functions on the manifold, while the Laplacian regularization only requests the function values for connected nodes to be as close as possible.

4 Semi-supervised SVMs

In this section, we first introduce the model of TiSVMs. Then we transform the optimization problem to a quadratic programming problem. Finally, we give the computational complexity and algorithmic description of TiSVMs.

4.1 TiSVMs

The SVM is a powerful tool for classification and regression. It is based on structural risk minimization that minimizes the upper bound of the generalization error [28]. The standard SVMs solve a quadratic programming problem and output the hyperplane that maximizes the margin between two parallel hyperplanes.

For the semi-supervised classification problem, suppose we have ℓ labeled examples and u unlabeled examples $\{\mathbf{x}_i\}_{i=1}^{\ell+u}$, and without loss of generalization the first ℓ examples correspond to the labeled ones with labels $y_i \in \{+1, -1\}$ $(i = 1, \ldots, \ell)$. Only binary classification is considered in this paper.

We propose a new method named tangent space intrinsic manifold regularized SVMs (TiSVMs) for semi-supervised learning, which attempts to solve the following problem

$$\begin{aligned} \min_{\{b_i, \mathbf{w}_i\}_{i=1}^{\ell+u}} \; & \frac{1}{\ell} \sum_{i=1}^{\ell} (1 - y_i f(\mathbf{x}_i))_+ + \gamma_1 \sum_{i=1}^{\ell+u} \|\mathbf{w}_i\|_2^2 \\ & + \gamma_2 R(\{b_i, \mathbf{w}_i\}_{i=1}^{\ell+u}), \end{aligned} \tag{8}$$

where $(1 - y_i f(\mathbf{x}_i))_+$ is the hinge loss [16], [26] defined as

$$(1 - y f(\mathbf{x}))_+ = \begin{cases} 1 - y f(\mathbf{x}), & \text{for } 1 - y f(\mathbf{x}) \geq 0 \\ 0, & \text{otherwise,} \end{cases}$$

$\gamma_1, \gamma_2 \geq 0$ are regularization coefficients, and $R(\{b_i, \mathbf{w}_i\}_{i=1}^{\ell+u})$ is the regularizer analogical to that given in (6). Note that, for labeled examples, $f(\mathbf{x}_i)$ is equal to the corresponding b_i.

For classification purpose, the classifier outputs for the unlabeled training examples are

$$y_i = \operatorname{sgn}(b_i), \quad i = \ell + 1, \ldots, \ell + u, \tag{9}$$

where $\operatorname{sgn}(\cdot)$ is the sign function which outputs 1 if its inputs are nonnegative and -1 otherwise. For a test example $\mathbf{x}$ which is not in $\{\mathbf{x}_i\}_{i=1}^{\ell+u}$, the SVM classifier predicts its label by

$$\operatorname{sgn}\Big(b_{\mathbf{z}(\mathbf{x})} + \mathbf{w}_{\mathbf{z}(\mathbf{x})}^\top T_{\mathbf{z}(\mathbf{x})} \big(\mathbf{x} - \mathbf{z}(\mathbf{x})\big)\Big), \tag{10}$$

where $\mathbf{z}(\mathbf{x}) = \arg\min_{\mathbf{z} \in \{\mathbf{x}_i\}_{i=1}^{\ell+u}} \|\mathbf{x} - \mathbf{z}\|_2$.

4.2 Optimization via Quadratic Programming

We now show that the optimization of (8) can be implemented by a standard quadratic programming solver. To begin with, using slack variables to replace the hinge loss, we rewrite (8) as

$$\begin{aligned} \min_{\{b_i, \mathbf{w}_i\}_{i=1}^{\ell+u}, \{\xi_i\}_{i=1}^{\ell}} \quad & \frac{1}{\ell} \sum_{i=1}^{\ell} \xi_i + \gamma_1 \sum_{i=1}^{\ell+u} \|\mathbf{w}_i\|_2^2 \\ & + \gamma_2 R(\{b_i, \mathbf{w}_i\}_{i=1}^{\ell+u}) \\ \text{s.t.} \quad & \begin{cases} y_i b_i \geq 1 - \xi_i, & i = 1, \ldots, \ell \\ \xi_i \geq 0, & i = 1, \ldots, \ell \,. \end{cases} \end{aligned} \tag{11}$$

Define $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_\ell)^\top$, and $\mathbf{a} = (\boldsymbol{\xi}^\top, \mathbf{b}^\top, \mathbf{w}^\top)^\top$. Suppose the first ℓ entries of $\mathbf{b}$ correspond to the ℓ labeled example $\mathbf{x}_1, \ldots, \mathbf{x}_\ell$, respectively. We can reformulate (11) as a standard quadratic program [27]

$$\begin{aligned} \min_{\mathbf{a}} \quad & \frac{1}{2} \mathbf{a}^\top H_1 \mathbf{a} + \mathbf{h}_1^\top \mathbf{a} \\ \text{s.t.} \quad & H_2 \mathbf{a} \leq \mathbf{h}_2 \,, \end{aligned} \tag{12}$$

where H_1 is a sparse matrix whose nonzero entries are only included in the bottom right sub-block H_1^{br} sized $(\ell + u)(m + 1) \times (\ell + u)(m + 1)$, $\mathbf{h}_1^\top = (\frac{1}{\ell}, \ldots, \frac{1}{\ell}, 0, \ldots, 0)$ with the first ℓ entries being nonzero, H_2 is a sparse matrix with 2ℓ rows, $\mathbf{h}_2^\top = (-1,\dots,-1,0,\dots,0)$ with the front ℓ entries being -1 and the back ℓ entries being 0. The matrix H_1^{br} is given by $2(\gamma_1 H_d + \gamma_2 S)$ where H_d is a diagonal matrix whose first $\ell+u$ diagonal elements are zero and the last $m(\ell+u)$ diagonal elements are 1, and S is taken from (7) where we set $k=\ell+u$ and $\mathbf{z}_i = \mathbf{x}_i$ $(i=1,\dots,k)$. The nonzero entries of H_2 can be specified as follows. For row i $(i \le \ell)$, the i-th element is -1 and the $(\ell+i)$-th element is $-y_i$. For row $\ell+i$ $(i\le\ell)$, the i-th element is -1.

TiSVMs solve a quadratic program and have the computational complexity of $O([(\ell+u)(m+1)+\ell]^3)$. For clarity, we explicitly state our tangent space intrinsic manifold regularized support vector machines algorithm in Algorithm 1.

Algorithm 1 Tangent Space Intrinsic Manifold Regularized Support Vector Machines (TiSVMs)

```
Input: $\ell$ labeled examples, $u$ unlabeled examples.
Obtain $H_1$, $h_1$, $H_2$, $h_2$.
Solve the quadratic programming (12) by using cross-validation to choose parameters.
Output: Predict the label of unlabeled training examples according to (9); predict the label of a new example according to (10).
```

5 Semi-Supervised TSVMs

In this section, we first introduce the model of TiTSVMs. Then we transform the optimization problems to quadratic programming problems. We give the computational complexity and algorithmic description of TiTSVMs. Finally, we introduce related work and make comparisons.

5.1 TiTSVMs

The twin support vector machine (TSVM) [17] is a nonparallel hyperplane classifier which aims to generate two nonparallel hyperplanes such that one of the hyperplanes is closer to one class and has a certain distance to the other class. Two classification hyperplanes are obtained by solving a pair of quadratic programming problems. The label of a new example is determined by the minimum of the perpendicular distances of the example to the two classification hyperplanes.

For the semi-supervised classification problem of TSVMs, suppose we have ℓ labeled examples which contain ℓ_1 positive examples, ℓ_2 negative examples and u unlabeled examples.

We propose a new method named TiTSVMs for semi-supervised learning, which attempts to solve the two quadratic programming problems in turn

$$\begin{aligned} \min_{\{b_i^+,\mathbf{w}_i^+\}_{i=1}^{\ell+u},\{\xi_i\}_{i=1}^{\ell_2}} \quad & \frac{1}{\ell_2}\sum_{i=1}^{\ell_2}\xi_i + \gamma_1\sum_{i=1}^{\ell_1}\|b_i^+\|_2^2 \\ & + \gamma_2 R(\{b_i^+,\mathbf{w}_i^+\}_{i=1}^{\ell+u}) \\ \text{s.t.} \quad & \begin{cases} b_{\ell_1+i}^+ \ge 1-\xi_i, & i=1,\dots,\ell_2 \\ \xi_i \ge 0, & i=1,\dots,\ell_2, \end{cases} \end{aligned} \tag{13}$$

and

$$\begin{aligned} \min_{\{b_i^-,\mathbf{w}_i^-\}_{i=1}^{\ell+u},\{\eta_i\}_{i=1}^{\ell_1}} \quad & \frac{1}{\ell_1}\sum_{i=1}^{\ell_1}\eta_i + \gamma_1\sum_{i=1}^{\ell_2}\|b_i^-\|_2^2 \\ & + \gamma_2 R(\{b_i^-,\mathbf{w}_i^-\}_{i=1}^{\ell+u}) \\ \text{s.t.} \quad & \begin{cases} b_{\ell_2+i}^- \ge 1-\eta_i, & i=1,\dots,\ell_1 \\ \eta_i \ge 0, & i=1,\dots,\ell_1 \\ b_i^- < b_{\ell_1+i}^+, & i=1,\dots,\ell_2 \\ b_{\ell_2+i}^- > b_i^+, & i=1,\dots,\ell_1, \end{cases} \end{aligned} \tag{14}$$

where $\gamma_1,\gamma_2 \ge 0$ are regularization coefficients, and $R(\{b_i^+,\mathbf{w}_i^+\}_{i=1}^{\ell+u})$ and $R(\{b_i^-,\mathbf{w}_i^-\}_{i=1}^{\ell+u})$ are the regularizer analogical to that given in (6). Note that, for labeled examples, $f^+(\mathbf{x}_i)$ is equal to the corresponding b_i^+ and $f^-(\mathbf{x}_i)$ is equal to the corresponding b_i^-. From (13), we obtain a set of classification hyperplanes such that positive examples are closer to them and negative examples are at a certain distance to them. Then, after we obtain the first set of classification hyperplanes, we add two additive constraints such that the distance of positive examples to the first set of classification hyperplanes is smaller than the distance to the other set of classification hyperplanes and the distance of negative example to the first set of classification hyperplanes is larger than the distance to the other.

For classification purpose, we at first search the nearest neighbor of a new example and find the tangent space representation of the nearest neighbor. The classifier outputs for the unlabeled training examples are

$$y_i = \text{sgn}(|b_i^-| - |b_i^+|), \quad i=\ell+1,\dots,\ell+u. \tag{15}$$

For a test example $\mathbf{x}$ which is not in $\{\mathbf{x}_i\}_{i=1}^{\ell+u}$, the TiTSVM classifier predicts its label by

$$\begin{aligned} \text{sgn}\Big(&|b_{\mathbf{z}(\mathbf{x})}^- + \mathbf{w}_{\mathbf{z}(\mathbf{x})}^{-\top} T_{\mathbf{z}(\mathbf{x})}(\mathbf{x}-\mathbf{z}(\mathbf{x}))| - |b_{\mathbf{z}(\mathbf{x})}^+ + \\ & \mathbf{w}_{\mathbf{z}(\mathbf{x})}^{+\top} T_{\mathbf{z}(\mathbf{x})}(\mathbf{x}-\mathbf{z}(\mathbf{x}))|\Big), \end{aligned} \tag{16}$$

where $\mathbf{z}(\mathbf{x}) = \arg\min_{\mathbf{z}\in\{\mathbf{x}_i\}_{i=1}^{\ell+u}} \|\mathbf{x}-\mathbf{z}\|_2$.

5.2 Optimization via Quadratic Programming

Define $\boldsymbol{\xi} = (\xi_1,\dots,\xi_{\ell_2})^\top$, $\boldsymbol{\eta} = (\eta_1,\dots,\eta_{\ell_1})^\top$, $\mathbf{a}^+ = (\boldsymbol{\xi}^\top, \mathbf{b}^{+\top}, \mathbf{w}^{+\top})^\top$, and $\mathbf{a}^- = (\boldsymbol{\eta}^\top, \mathbf{b}^{-\top}, \mathbf{w}^{-\top})^\top$. Suppose the first ℓ entries of $\mathbf{b}^+$ and $\mathbf{b}^-$ correspond to the ℓ labeled example $\mathbf{x}_1,\dots,\mathbf{x}_\ell$, respectively. We can reformulate (13) as a standard quadratic program

$$\begin{aligned} \min_{\mathbf{a}^+} \quad & \frac{1}{2}\mathbf{a}^{+\top} H_1^+ \mathbf{a}^+ + \mathbf{h}_1^{+\top}\mathbf{a}^+ \\ \text{s.t.} \quad & H_2^+\mathbf{a}^+ \le \mathbf{h}_2^+ , \end{aligned} \tag{17}$$

where H_1^+ is a sparse matrix whose nonzero entries are only included in the bottom right sub-block H_1^{+br} sized $(\ell+u)(m+1)\times(\ell+u)(m+1)$, $\mathbf{h}_1^{+\top} = (\frac{1}{\ell_2},\dots,\frac{1}{\ell_2},0,\dots,0)$ with the first ℓ_2 entries being nonzero, H_2^+ is a sparse matrix with $2\ell_2$ rows, and $\mathbf{h}_2^{+\top} = (-1,\dots,-1,0,\dots,0)$ with the front ℓ_2 entries being -1 and the back ℓ_2 entries being 0. The matrix H_1^{+br} is given by $2\gamma_2 S^+ + M_{\ell_1}$ where S^+ is taken from (7) by setting $k=\ell+u$ and $\mathbf{z}_i = \mathbf{x}_i$ $(i = 1,\dots,k)$ and M_{ℓ_1} is a diagonal matrix with the first ℓ_1 diagonal elements being $2\gamma_1$ and the rest 0. The nonzero entries of H_2^+ can be specified as follows. For row i $(i \le \ell_2)$, both the i-th element and the $(\ell_1+\ell_2+i)$-th element are -1. For row ℓ_2+i $(i\le\ell_2)$, the i-th element is -1.

Then we reformulate (14) as a standard quadratic program

$$\begin{aligned} \min_{\mathbf{a}^-} \quad & \frac{1}{2}\mathbf{a}^{-\top} H_1^- \mathbf{a}^- + \mathbf{h}_1^{-\top}\mathbf{a}^- \\ \text{s.t.} \quad & H_2^- \mathbf{a}^- \le \mathbf{h}_2^- \ , \end{aligned} \tag{18}$$

where H_1^- is a sparse matrix whose nonzero entries are only included in the bottom right sub-block H_1^{-br} sized $(\ell+u)(m+1)\times(\ell+u)(m+1)$, $\mathbf{h}_1^{-\top} = (\frac{1}{\ell_1},\dots,\frac{1}{\ell_1},0,\dots,0)$ with the first ℓ_1 entries being nonzero, H_2^- is a sparse matrix with $2\ell_1+\ell_2+\ell_1$ rows, and $\mathbf{h}_2^{-\top} = (-1,\dots,-1,0,\dots,0,\dots,b^+_{\ell_1+1},\dots,b^+_{\ell_1+\ell_2},-b^+_1,\dots,-b^+_{\ell_1})$ with the front ℓ_1 entries being -1, the back ℓ_1 entries being 0. The matrix H_1^{-br} is given by $2\gamma_2 S^- + M_{\ell_2}$ where S^- is taken from (7) by setting $k=\ell+u$ and $\mathbf{z}_i=\mathbf{x}_i$ $(i=1,\dots,k)$ and M_{ℓ_2} is a diagonal matrix with the first ℓ_2 diagonal elements being $2\gamma_1$ and the rest 0. The nonzero entries of H_2^- can be specified as follows. For row i $(i\le\ell_1)$, both the i-th element and the $(\ell_1+\ell_2+i)$-th element are -1. For row ℓ_1+i $(i\le\ell_1)$, the i-th element is -1. For row $2\ell_1+i$ $(i\le\ell_2)$, the (ℓ_1+i)-th element is 1. For row $2\ell_1+\ell_2+i$ $(i\le\ell_1)$, the $(\ell_1+\ell_2+i)$-th element is -1.

TiTSVMs solve a pair of quadratic programs and have the computational complexity of $O([(\ell+u)(m+1)+\ell_1]^3 + [(\ell+u)(m+1)+\ell_2]^3)$. For clarity, we explicitly state our tangent space intrinsic manifold regularized twin support vector machines algorithm in Algorithm 2.

Algorithm 2 Tangent Space Intrinsic Manifold Regularized Twin Support Vector Machines (TiTSVMs)

```
1: Input: $\ell$ labeled examples ($\ell_1$ positive examples and $\ell_2$ negative examples), $u$ unlabeled examples.
2: Obtain $H_1^+$, $H_1^-$, $h_1^+$, $h_1^-$, $H_2^+$, $H_2^-$, $h_2^+$, $h_2^-$.
3: Solve the quadratic programming (17) and (18) by using cross-validation to choose parameters.
4: Output: Predict the label of unlabeled training examples according to (15); predict the label of a new example according to (16).
```

5.3 Comparisons with Related Work

For the purpose of comparison, below we give the objective functions for SVMs [16], [28], TSVMs [17] and LapSVMs [3], [10], respectively.

In its most natural form, the optimization of soft-margin SVMs for supervised classification is to find a function f_s from a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$. Optimization in the original space is subsumed in the RKHS case with a linear kernel. Given a set of ℓ labeled examples $\{(\mathbf{x}_i, y_i)\}$ $(i=1,...,\ell)$ with $y_i \in \{+1,-1\}$, the optimization problem is

$$\min_{f_s\in\mathcal{H}} \frac{1}{\ell}\sum_{i=1}^{\ell}(1-y_i f_s(\mathbf{x}_i))_+ + \gamma_s\|f_s\|_2^2,$$

where $\gamma_s \ge 0$ is a norm regularization parameter.

Suppose examples belonging to classes 1 and -1 are represented by matrices A_+ and B_-, and the size of A_+ and B_- are $(\ell_1\times d)$ and $(\ell_2\times d)$, respectively. We define two matrices A, B and four vectors $\mathbf{v}_1$, $\mathbf{v}_2$, $\mathbf{e}_1$, $\mathbf{e}_2$, where $\mathbf{e}_1$ and $\mathbf{e}_2$ are vectors of ones of appropriate dimensions and

$$A=(A_+,\mathbf{e}_1),\ B=(B_-,\mathbf{e}_2),\ \mathbf{v}_1=\begin{pmatrix}\mathbf{w}_1\\ b_1\end{pmatrix},\ \mathbf{v}_2=\begin{pmatrix}\mathbf{w}_2\\ b_2\end{pmatrix}. \tag{19}$$

TSVMs obtain two nonparallel hyperplanes

$$\mathbf{w}_1^\top\mathbf{x}+b_1=0 \quad\text{and}\quad \mathbf{w}_2^\top\mathbf{x}+b_2=0 \tag{20}$$

around which the examples of the corresponding class get clustered. The classifier is given by solving the following quadratic programs separately

(TSVM1)

$$\begin{aligned} \min_{\mathbf{v}_1,\mathbf{q}_1} \quad & \frac{1}{2}(A\mathbf{v}_1)^\top(A\mathbf{v}_1)+c_1\mathbf{e}_2^\top\mathbf{q}_1 \\ \text{s.t.} \quad & -B\mathbf{v}_1+\mathbf{q}_1\succeq\mathbf{e}_2,\ \mathbf{q}_1\succeq 0, \end{aligned} \tag{21}$$

(TSVM2)

$$\begin{aligned} \min_{\mathbf{v}_2,\mathbf{q}_2} \quad & \frac{1}{2}(B\mathbf{v}_2)^\top(B\mathbf{v}_2)+c_2\mathbf{e}_1^\top\mathbf{q}_2 \\ \text{s.t.} \quad & A\mathbf{v}_2+\mathbf{q}_2\succeq\mathbf{e}_1,\ \mathbf{q}_2\succeq 0, \end{aligned} \tag{22}$$

where c_1, c_2 are nonnegative parameters and $\mathbf{q}_1$, $\mathbf{q}_2$ are slack vectors of appropriate dimensions. The label of a new example $\mathbf{x}$ is determined by the minimum of $|\mathbf{x}^\top\mathbf{w}_r+b_r|$ $(r=1,2)$ which are the perpendicular distances of $\mathbf{x}$ to the two hyperplanes given in (20).

The intension of LapSVMs is for effective semi-supervised learning under the local smoothness regularization. The role of unlabeled data is to restrict the capacity of the considered function set through the Laplacian matrix L. Namely, desirable functions must be smooth across all the training examples. Given ℓ labeled and u unlabeled examples, LapSVMs solve the following problem in an RKHS

$$\min_{f_{ls}\in\mathcal{H}} \frac{1}{\ell}\sum_{i=1}^{\ell}(1-y_i f_{ls}(\mathbf{x}_i))_+ + \gamma_1\|f_{ls}\|_2^2+\gamma_2\mathbf{f}_{ls}{}^\top L\mathbf{f}_{ls},$$

where γ_1 and γ_2 are nonnegative regularization parameters as given before, and vector $\mathbf{f}_{ls} = (f_{ls}(\mathbf{x}_1),\dots,f_{ls}(\mathbf{x}_{\ell+u}))^\top$.

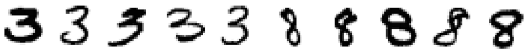

Fig. 1. Examples of digits 3 and 8 in the handwritten digit data set.

Fig. 2. Examples of face and non-face images in the face detection data set.

SVMs, TSVMs and LapSVMs are important learning machines making use of the "kernel trick" where a linear classification function in a potentially high-dimensional RKHS is learned, whose relationship with respect to the original inputs is usually nonlinear when nonlinear kernel maps are adopted to generate the space. There are two main differences between them and our TiSVMs and TiTSVMs: i) They seek linear functions in probably high-dimensional kernel spaces [19], while TiSVMs and TiTSVMs pursue linear functions in low-dimensional tangent spaces of manifolds; ii) They only learn one single classification function, while TiSVMs and TiTSVMs learn many $(b_i, \mathbf{w}_i)$ pairs which are then dynamically selected to determine the label of some test example depending on the distances between this example and the training examples. However, there is a common property among all the considered learning machines. That is, they are all able to learn nonlinear classifiers with respect to the original feature representation.

6 Experiments

We evaluated the tangent space intrinsic manifold regularization for semi-supervised classification with multiple real-world datasets. There is a parameter γ, which is intrinsic in the regularization matrix S. This parameter leverages the two components in (6). Because there are not very reasonable and necessary reasons to overweight one over the other, we treat them equally in S, namely $\gamma = 1$. Other parameters in classifiers are selected through cross-validation. For example, regularization parameters for TiSVMs and TiTSVMs, namely γ_1 and γ_2, are selected from the set $\{10^{-6}, 10^{-4}, 10^{-2}, 1, 10, 100\}$ through a two-fold cross-validation of labeled training examples on the training set.

6.1 Handwritten Digit Classification

This dataset comes from the UCI Machine Learning Repository. The data we use here include 2400 examples of digits 3 and 8 chosen from the MNIST digital images[1], and half of the data are digit 3. Image sizes are 28×28. Ten images are shown in Figure 1. The training set includes 80% of the data, and the remaining 20% serve as the test set. For semi-supervised classification, 1% of the whole data set are randomly selected from the training set to serve as labeled training data.

For the construction of adjacency graphs, 10 nearest neighbors are used. For edge weight assignment, we choose the polynomial kernel of degree 3, following [30] and [3] on similar classification tasks. The local tangent spaces are fixed to be 2-dimensional and 3-dimensional, since for handwritten digits people usually obtain good embedding results with these dimensionalities.

With the identified regularization parameters, we retrain TiSVMs and TiTSVMs, and evaluate performances on the test data and unlabeled training data, respectively. The entire process is repeated over 10 random divisions of the training and test sets, and the reported performance is the averaged accuracy and the corresponding standard deviation. Besides SVMs, LapSVMs and LapTSVMs, parallel field regularization (PFR) [29] is also adopted to compare with our method under the same setting, e.g., the same kernel is used, and regularization parameters are selected with the same two-fold cross-validation method from the same range. PFR, which is briefly mentioned in Section 7.4, is a recent semi-supervised regression method which is used here for classification. For SVMs the unlabeled training data are neglected because they are only for semi-supervised learning.

6.2 Face Detection

Face detection is a binary classification problem which intends to identify whether a picture is a human face or not. In this experiment, 2000 face and non-face images from the MIT CBCL repository[2] [31], [32] are used, where half of them are faces and each image is a 19×19 gray picture. Figure 2 shows a number of examples. The same experimental setting with the previous handwritten digit classification is adopted, such as the percentage of training data and the percentage of labeled data.

6.3 Speech Recognition

The Isolet dataset[3] comes from the UCI Machine Learning Repository. It is collected from 150 subjects speaking each letter of the alphabet twice. Hence, we have 52 training examples from each speaker. Due to the lack of three examples, there are 7797 examples in total. These speakers are grouped into five sets of 30 speakers each. These groups are referred to as isolet1-isolet5. Each of these datasets has 26 classes. The attribute information include spectral coefficients, contour features, sonorant features, pre-sonorant features and post-sonorant features. In Isolet dataset, we choose two classes (a, b) for classification. So there are in total 480 examples. The training set includes 80% of the data, and the remaining 20% serve as the test set. For semi-supervised classification, $1/48$ of the whole data set are randomly

[1] http://yann.lecun.com/exdb/mnist/

[2] http://cbcl.mit.edu/software-datasets/FaceData2.html

[3] http://archive.ics.uci.edu/ml/datasets/

TABLE 1
Classification Accuracies (%) (with Standard Deviations) and Time (with Standard Deviations) of Different Methods on the Handwritten Digit Classification Data.

	SVM	LapSVM	PFR(m=2)	TiSVM(m=2)
$\mathbb{U}$	88.04 (1.78)	90.56 (1.74)	88.51 (6.64)	92.09 (1.58)
$\mathbb{T}$	87.98 (1.96)	90.40 (1.37)	88.56 (6.25)	**92.40** (1.79)
$T1$	$3.62\times10^{-1}(4.86\times10^{-2})$	3.03×10^{2} (9.75)	6.05×10^{3} (7.07×10)	6.70×10^{3} (1.6×10^{2})
$T2$	6.87×10^{-2} (2.11×10^{-2})	5.90×10^{-3} (1.40×10^{-3})	3.00×10^{-4} (4.80×10^{-4})	3.00×10^{-4} (4.80×10^{-4})
$T3$	1.55×10^{-2} (5.00×10^{-4})	1.55×10^{-1} (2.39×10^{-2})	1.57×10^{-1} (2.24×10^{-2})	1.55×10^{-1} (1.28×10^{-2})

	TiTSVM(m=2)	PFR(m=3)	TiSVM(m=3)	TiTSVM(m=3)	LapTSVM
$\mathbb{U}$	91.21 (1.99)	90.48 (2.63)	91.49 (1.88)	90.43 (2.52)	**92.80** (1.71)
$\mathbb{T}$	91.48 (2.34)	90.73 (2.87)	91.90 (1.85)	90.71 (2.29)	90.31 (2.37)
$T1$	4.52×10^{4} (8.26×10^{2})	7.29×10^{3} (2.13×10^{2})	8.95×10^{3} (2.08×10^{2})	1.24×10^{5} (9.31×10^{3})	2.88×10^{3} (1.96×10^{2})
$T2$	2.00×10^{-3} (0.00)	4.00×10^{-4} (5.16×10^{-4})	4.00×10^{-4} (5.16×10^{-4})	4.50×10^{-3} (4.30×10^{-3})	1.03×10 (8.45×10^{-1})
$T3$	2.22×10^{-1} (8.74×10^{-2})	1.89×10^{-1} (7.00×10^{-2})	1.85×10^{-1} (4.61×10^{-2})	1.96 (1.96)	2.59 (3.16×10^{-1})

TABLE 2
Classification Accuracies (%) (with Standard Deviations) and Time (with Standard Deviations) of Different Methods on the Face Detection Data.

	SVM	LapSVM	PFR(m=2)	TiSVM(m=2)
$\mathbb{U}$	76.42 (6.34)	79.42 (6.65)	80.25 (7.21)	**85.42** (3.86)
$\mathbb{T}$	76.43 (6.12)	79.43 (5.65)	79.75 (7.76)	**84.63** (3.64)
$T1$	3.58×10^{-1} (4.97×10^{-2})	1.87×10^{2} (4.83)	1.97×10^{3} (1.29×10)	2.27×10^{3} (6.64×10)
$T2$	2.12×10^{-2} (2.70×10^{-3})	3.90×10^{-3} (8.75×10^{-4})	1.00×10^{-4} (3.16×10^{-4})	1.00×10^{-4} (3.16×10^{-4})
$T3$	5.50×10^{-3} (1.40×10^{-3})	7.70×10^{-2} (9.30×10^{-3})	7.21×10^{-2} (9.70×10^{-3})	6.40×10^{-2} (3.30×10^{-3})

	TiTSVM(m=2)	PFR(m=3)	TiSVM(m=3)	TiTSVM(m=3)	LapTSVM
$\mathbb{U}$	77.04 (10.78)	80.17 (7.77)	85.26 (4.11)	77.94 (7.79)	81.44 (2.77)
$\mathbb{T}$	77.22 (10.73)	79.95 (7.95)	84.58 (3.97)	78.70 (8.40)	81.12 (3.26)
$T1$	2.68×10^{4} (5.48×10^{2})	2.89×10^{3} (8.43×10)	4.17×10^{3} (1.13×10^{2})	6.69×10^{4} (7.86×10^{3})	1.77×10^{3} (2.48×10)
$T2$	1.90×10^{3} (3.16×10^{-4})	2.00×10^{-4} (4.22×10^{-4})	2.00×10^{-4} (4.22×10^{-4})	2.90×10^{-3} (3.20×10^{-3})	6.34 (3.00×10^{-1})
$T3$	7.75×10^{-2} (1.82×10^{-2})	6.53×10^{-2} (1.19×10^{-2})	7.39×10^{-2} (9.70×10^{-3})	4.00×10^{-1} (5.24×10^{-1})	1.56 (5.13×10^{-2})

selected from the training set to serve as labeled training examples. The same experimental setting is adopted as the previous experiments.

6.4 German Credit Data

This German Credit dataset[4] also comes from the UCI Machine Learning Repository and consists of 1000 examples (300 positive examples and 700 negative examples). The training set includes 80% of the data, and the remaining 20% serve as the test set. For semi-supervised classification, 1% of the whole data set are randomly selected from the training set to serve as the labeled training data. The same experimental setting is adopted as the previous experiments.

6.5 Australian

The Australian dataset[5] contains 690 examples (307 positive examples and 383 negative examples) and 14 attributes. The training set includes 80% of the data, and the remaining 20% serve as the test set. For semi-supervised classification, $1/30$ of the whole data set are randomly selected from the training set to serve as the labeled training data. In this dataset, we use RBF kernel and set the adjusting kernel parameter to be 100 which can perform well for LapSVMs. The other experimental setting is adopted as the previous experiments.

6.6 Contraceptive Method Choice

The Contraceptive Method Choice dataset[6] contains 1473 examples containing three classes. We choose 629 positive examples and 511 negative examples for binary classification. The training set includes 80% of the data, and the remaining 20% serve as the test set. For semi-supervised classification, $1/50$ of the whole data set are randomly selected from the training set to serve as the labeled training data. In this dataset, we use RBF kernel and set the adjusting kernel parameter to be 100 which can perform well for LapSVMs. The other experimental setting is adopted as the previous experiments.

6.7 Experimental Results

Handwritten Digit Classification: The classification accuracies and time of different methods on this dataset are

[4] https://archive.ics.uci.edu/ml/datasets/Statlog+(German+Credit+Data)

[5] http://archive.ics.uci.edu/ml/datasets/Statlog+%28Australian+Credit+Approval%29

[6] https://archive.ics.uci.edu/ml/datasets/Contraceptive+Method+Choice

TABLE 3
Classification Accuracies (%) (with Standard Deviations) and Time (with Standard Deviations) of Different Methods on the Isolet Data.

	SVM	LapSVM	PFR(m=2)	TiSVM(m=2)
$\mathbb{U}$	93.93 (3.30)	97.11 (0.95)	95.61 (8.01)	97.49 (0.63)
$\mathbb{T}$	94.16 (4.48)	97.08 (2.90)	95.42 (9.82)	98.23 (0.85)
$T1$	4.20×10^{-2} (7.40×10^{-3})	4.54 (2.79×10^{-1})	5.22×10^{2} (2.01×10)	3.54×10^{2} (7.00)
$T2$	1.51×10^{-2} (3.16×10^{-2})	5.00×10^{-4} (5.30×10^{-4})	1.00×10^{-4} (3.16×10^{-4})	1.00×10^{-4} (3.16×10^{-4})
$T3$	1.30×10^{-2} (6.75×10^{-4})	8.60×10^{-3} (2.40×10^{-3})	1.98×10^{-2} (2.48×10^{-2})	9.90×10^{-3} (2.10×10^{-3})

	TiTSVM(m=2)	PFR(m=3)	TiSVM(m=3)	TiTSVM(m=3)	LapTSVM
$\mathbb{U}$	**98.18** (0.92)	95.24 (8.53)	97.27 (0.47)	97.41 (1.78)	97.91 (0.85)
$\mathbb{T}$	**98.65** (1.21)	95.42 (10.57)	97.81 (1.04)	98.23 (1.63)	98.33 (1.22)
$T1$	1.08×10^{3} (1.06×10^{2})	3.97×10^{2} (1.72×10)	4.25×10^{2} (2.10×10)	1.67×10^{3} (7.99×10)	9.30×10 (4.82)
$T2$	5.00×10^{-4} (5.27×10^{-4})	1.00×10^{-4} (4.25×10^{-4})	1.00×10^{-4} ($3.164.25\times10^{-4}$)	7.00×10^{-4} (4.83×10^{-4})	8.26×10^{-2} (1.47×10^{-2})
$T3$	1.26×10^{-2} (3.50×10^{-3})	1.25×10^{-2} (4.50×10^{-3})	1.04×10^{-2} (2.40×10^{-3})	1.15×10^{-2} (2.20×10^{-3})	2.48×10^{-2} (8.00×10^{-3})

TABLE 4
Classification Accuracies (%) (with Standard Deviations) and Time (with Standard Deviations) of Different Methods on the German Credit Data.

	SVM	LapSVM	PFR(m=2)	TiSVM(m=2)
$\mathbb{U}$	60.03 (7.63)	67.48 (4.22)	**70.00** (0.00)	**70.00** (0.00)
$\mathbb{T}$	59.50 (6.21)	66.55 (3.47)	**70.00** (0.00)	**70.00** (0.00)
$T1$	1.05×10^{-1} (1.35×10^{-1})	2.79×10 (1.86)	3.28×10^{2} (8.32)	6.90×10^{2} (2.58×10^{2})
$T2$	1.10×10^{-3} (1.10×10^{-3})	1.10×10^{-3} (3.16×10^{-4})	1.00×10^{-4} (3.16×10^{-4})	1.00×10^{-4} (3.16×10^{-4})
$T3$	4.00×10^{-4} (6.99×10^{-4})	1.74×10^{-2} (3.10×10^{-3})	1.63×10^{-2} (1.01×10^{-2})	2.04×10^{-2} (1.96×10^{-2})

	TiTSVM(m=2)	PFR(m=3)	TiSVM(m=3)	TiTSVM(m=3)	LapTSVM
$\mathbb{U}$	66.55 (0.92)	**70.00** (0.00)	**70.00** (0.00)	68.13 (3.79)	61.96 (7.44)
$\mathbb{T}$	67.55 (4.46)	**70.00** (0.00)	**70.00** (0.00)	67.10 (5.11)	62.30 (7.26)
$T1$	4.42×10^{3} (8.26×10)	4.14×10^{2} (8.21)	4.92×10^{2} (1.35×10)	9.18×10^{3} (4.03×10^{2})	3.40×10^{2} (1.08×10)
$T2$	1.00×10^{-3} (2.12×10^{-5}) (8.21)	1.00×10^{-4} (3.16×10^{-4})	1.00×10^{-4} (3.16×10^{-4})	1.00×10^{-3} (0.00)	7.42×10^{-1} (6.71×10^{-2})
$T3$	1.14×10^{-2} (2.30×10^{-3})	1.16×10^{-2} (3.50×10^{-3})	1.06×10^{-2} (2.90×10^{-3})	2.5610^{-2} (4.83×10^{-2})	1.87×10^{-1} (2.87×10^{-2})

shown in Table 1, where $\mathbb{U}$ and $\mathbb{T}$ represent the accuracy on the unlabeled training and test data, respectively, and the best accuracies are indicated in bold. $T1$, $T2$ and $T3$ (all in seconds) represent training time, test time on the unlabeled data and test time on the labeled data, respectively. From this table, we see that semi-supervised methods give better performance than the supervised SVMs, which indicates the usefulness of unlabeled examples. Moreover, the proposed TiSVMs and TiTSVMs perform much better than LapSVMs and PFR. TiSVMs perform a little better than TiTSVMs. LapTSVMs perform best on the classification of the unlabeled data while TiSVMs performs best on the classification of the labeled data.

Face Detection: The classification results and time of TiSVMs and TiTSVMs with comparisons to other methods are given in Table 2, which show that our TiSVMs have got the best accuracies on both the test and unlabeled training data. However, the performance of our TiTSVMs is not good, just a little better than SVMs. We conjecture this is caused by the fact that TiTSVMs have more parameters to train and thus it is sometimes difficult to return a very good solution.

Speech Recognition: The classification accuracies and time of different methods on this dataset are shown in Table 3. This table shows that semi-supervised methods give better performance than the supervised SVMs. Moreover, the proposed TiSVMs and TiTSVMs perform much better than LapSVMs and PFR. LapTSVMs and TiTSVMs perform a little better than TiSVMs.

German Credit Data: The classification accuracies and time of different methods on this dataset are shown in Table 4. Again, we see that semi-supervised methods give better performance than the supervised SVMs. Moreover, the proposed TiSVMs perform much better than TiTSVMs, LapSVMs, LapTSVMs and SVMs, and perform the same as PFR. TiTSVMs perform a little better than LapSVMs on the classification of the labeled data and a little worse than LapSVMs on the classification of the unlabeled data.

Australian: The classification accuracies and time of different methods on this dataset are shown in Table 5. We see that semi-supervised methods give better performance than the supervised SVMs. Moreover, the proposed TiTSVMs perform much better than TiSVMs, LapSVMs, LapTSVMs and SVMs.

Contraceptive Method Choice: The classification accuracies and time of different methods on this dataset are shown in Table 6. Semi-supervised methods give better performance than the supervised SVMs. TiTSVMs ($m = 3$) perform best on the classification of the unlabeled data and TiSVMs ($m = 2$) perform best on the

TABLE 5
Classification Accuracies (%) (with Standard Deviations) and Time (with Standard Deviations) of Different Methods on the Australian Data.

	SVM	LapSVM	PFR(m=2)	TiSVM(m=2)
$\mathbb{U}$	56.25 (5.92)	60.44 (4.76)	61.70 (4.92)	61.02 (4.52)
$\mathbb{T}$	54.86 (7.60)	60.14 (6.12)	61.16 (5.10)	60.58 (3.17)
$T1$	6.33×10^{-2} (9.98×10^{-2})	1.34×10 (9.56×10^{-1})	1.65×10^{2} (6.80)	2.03×10^{3} (7.16×10^{2})
$T2$	1.50×10^{-3} (1.30×10^{-3})	5.00×10^{-4} (5.27×10^{-4})	1.00×10^{-4} (3.16×10^{-4})	1.00×10^{-4} (3.16×10^{-4})
$T3$	9.00×10^{-4} (5.68×10^{-4})	1.74×10^{-2} (3.86×10^{-2})	6.70×10^{-3} (1.70×10^{-3})	6.10×10^{-3} (1.90×10^{-3})

	TiTSVM(m=2)	PFR(m=3)	TiSVM(m=3)	TiTSVM(m=3)	LapTSVM
$\mathbb{U}$	**64.64** (3.92)	62.20 (4.74)	62.10 (4.19)	62.36 (4.54)	59.96 (6.64)
$\mathbb{T}$	63.48 (6.49)	62.32 (6.74)	60.72 (4.19)	**64.04** (6.94)	60.80 (11.84)
$T1$	4.49×10^{3} (1.14×10^{3})	2.09×10^{2} (7.67)	2.84×10^{3} (7.47×10^{2})	1.05×10^{4} (2.16×10^{3})	1.64×10^{2} (1.68×10)
$T2$	8.00×10^{-4} (4.22×10^{-4}) (8.21)	1.40×10^{-3} (4.10×10^{-3})	1.40×10^{-3} (4.10×10^{-3})	8.75×10^{-4} (3.53×10^{-4})	1.70×10^{-1} (1.85×10^{-2})
$T3$	7.70×10^{-3} (3.20×10^{-3})	7.90×10^{-3} (4.00×10^{-3})	7.90×10^{-3} (4.40×10^{-3})	1.92×10^{-2} (3.27×10^{-2})	4.08×10^{-2} (7.60×10^{-3})

TABLE 6
Classification Accuracies (%) (with Standard Deviations) and Time (with Standard Deviations) of Different Methods on the Contraceptive Method Choice Data.

	SVM	LapSVM	PFR(m=2)	TiSVM(m=2)
$\mathbb{U}$	55.81 (1.00)	56.20 (3.47)	56.53 (1.29)	56.71 (1.19)
$\mathbb{T}$	55.13 (2.08)	**57.06** (3.93)	56.54 (3.48)	**57.06** (3.79)
$T1$	7.70×10^{-2} (1.02×10^{-2})	3.77×10 (1.37×10^{-1})	4.86×10^{2} (2.66×10)	5.03×10^{2} (9.39×10)
$T2$	1.10×10^{-3} (3.16×10^{-4})	1.40×10^{-3} (5.16×10^{-4})	3.00×10^{-4} (4.83×10^{-4})	3.00×10^{-4} (4.83×10^{-4})
$T3$	3.00×10^{-4} (4.83×10^{-4})	8.80×10^{-3} (6.32×10^{-4})	2.51×10^{-2} (1.09×10^{-2})	1.27×10^{-2} (2.20×10^{-3})

	TiTSVM(m=2)	PFR(m=3)	TiSVM(m=3)	TiTSVM(m=3)	LapTSVM
$\mathbb{U}$	56.58 (1.74)	56.90 (1.41)	56.57 (1.18)	**56.98** (0.02)	56.37 (6.05)
$\mathbb{T}$	56.14 (3.77)	56.89 (3.91)	56.71 (3.03)	56.62 (4.27)	56.93 (6.46)
$T1$	4.70×10^{3} (9.40×10^{2})	6.41×10^{2} (6.13)	5.60×10^{2} (2.54×10)	8.68×10^{3} (4.80×10^{2})	4.45×10^{2} (2.36×10)
$T2$	1.00×10^{-3} (0.00)	1.00×10^{-4} (3.16×10^{-4})	1.00×10^{-4} (3.16×10^{-4})	1.10×10^{-3} (3.16×10^{-4})	1.19 (6.95×10^{-2})
$T3$	1.84×10^{-2} (5.10×10^{-3})	2.30×10^{-2} (7.50×10^{-3})	1.89×10^{-2} (2.50×10^{-2})	2.02×10^{-2} (2.05×10^{-2})	3.10×10^{-1} (4.53×10^{-2})

TABLE 7
Classifier Rank of All the Methods

method	SVM	LapSVM	PFR(m=2)	TiSVM(m=2)	TiTSVM(m=2)	PFR(m=3)	TiSVM(m=3)	TiTSVM(m=3)	LapTSVM
Average rank ($\mathbb{U}$)	9.00	6.33	5.25	**2.83**	4.33	4.42	3.67	4.33	4.83
Average rank ($\mathbb{T}$)	9.00	5.75	5.67	**2.75**	4.50	4.17	3.75	4.75	4.67

classification of the labeled data.

Table 7 lists the average rank of all the methods for their classification accuracies, from which we can conclude our methods outperform other methods. From the perspective of time, as can be seen from Table 1∼6, they usually take more training time, less test time on the unlabeled data and more test time on the labeled data.

7 Discussions

Here we discuss some possible improvements and extensions of the proposed tangent space intrinsic manifold regularization and semi-supervised classification algorithms, which can be very helpful to adapt to different applications.

7.1 Out-of-Sample Extension Using Multiple Neighbors

In this paper, we only used one nearest neighbor from the training data to represent the out-of-sample extension for the learned functions, as given in (3). The performance would depend largely on the property of the selected neighboring point.

However, in order to enhance the robustness, we can adopt a weighted average to represent the function $f(\mathbf{x})$ for $\mathbf{x}$ out of the training set Z. Suppose we adopt n neighbors to carry out the out-of-example extension, and $\mathcal{N}(\mathbf{x}) \subset Z$ includes the n neighbors of $\mathbf{x}$. Then $f(\mathbf{x})$ is computed as follows

$$f(\mathbf{x}) = \frac{1}{\sum_{\mathbf{z}\in\mathcal{N}(\mathbf{x})} W_{\mathbf{xz}}} \sum_{\mathbf{z}\in\mathcal{N}(\mathbf{x})} W_{\mathbf{xz}} \big[b_{\mathbf{z}} + \mathbf{w}_{\mathbf{z}}^{\top} T_{\mathbf{z}} (\mathbf{x} - \mathbf{z}) \big],$$

where $W_{\mathbf{xz}}$ is the weight calculated with the same manner as constructing the weighted graphs from Z. Here n, which is not necessarily equal to the neighborhood number used to construct the adjacency graphs, can be selected through some appropriate model selection procedure. (10) and (16) can be extended similarly.

7.2 Reducing Anchor Points

We treated each example from the training set as an anchor point, where local PCA is used to calculate the tangent space. The number of parameters that should be estimated in our methods 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 anchor points where only "key" examples are kept as anchor points and the function values for the other examples are extended from those of the anchor points. This is a kind of research for data set sparsification. People can devise different methods to find the examples which they regard as "key".

The research of anchor point reduction is especially useful whether training data are very limited or of large-scale. For limited data, the precision for parameter estimation can be improved as a result of parameter reduction, while for large-scale data, anchor point reduction can be promising to speed up the training process. For example, reducing anchor points can be applied to semi-supervised learning where the training data include a large number of unlabeled examples.

7.3 Improving the Estimation of Tangent Spaces

For the manifold learning problems considered in this paper, the estimation of bases for tangent spaces is an important step where local PCA with fixed neighborhood size was used. This is certainly not the optimal choice, since data could be non-uniformly sampled and manifolds can have a varying curvature. Notice that the neighborhood size can determine the evolution of calculated tangent spaces along the manifold. If a small neighborhood size is used, the tangent spaces would change more sharply when the manifold is not flat. Moreover, noise 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 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 concern of the current approach.

7.4 Semi-Supervised Regression

The proposed tangent space intrinsic manifold regularization can also be applied to semi-supervised regression problems. Using the same notations as in Section 4, we give the following objective function for semi-supervised regression with the squared loss

$$\begin{aligned}\min_{\{b_i,\mathbf{w}_i\}_{i=1}^{\ell+u}} \ & \frac{1}{\ell}\sum_{i=1}^{\ell}\big(y_i - f(\mathbf{x}_i)\big)^2 + \gamma_1\sum_{i=1}^{\ell+u}\|\mathbf{w}_i\|_2^2 \\ & + \gamma_2 R(\{b_i,\mathbf{w}_i\}_{i=1}^{\ell+u}).\end{aligned} \qquad (23)$$

The generalization to examples not in the training set is analogical to (10) but without the sign function.

It can be easily shown that the optimization of (23) is obtained by solving a sparse linear system, which is much simpler than the quadratic programming for the classification case. This is reminiscent of the works by [33] and [29], which are both on semi-supervised regression preferring linear functions on manifolds. These works perform regression from the more complex perspectives of Hessian energy and parallel fields, respectively. There are two main differences between (23) and their works: i) Their objective functions do not contain the second term of (23) and thus cannot regularize the norms of $\{\mathbf{w}_i\}_{i=1}^{\ell+u}$; ii) Their works focus on transductive learning while by solving (23) we can do both inductive and transductive learning. Comparing the performances of these methods on regression is not the focus of this paper, which can be explored as interesting future work.

7.5 Entirely Supervised Learning

The TiSVMs for semi-supervised classification can be extended to the entirely supervised learning scenario when all the training examples are labeled (a similar extension of TiTSVMs is also possible). In this case, the objective function could probably have the following form

$$\begin{aligned}\min_{\{b_i,\mathbf{w}_i\}_{i=1}^{\ell}} \ & \frac{1}{\ell}\sum_{i=1}^{\ell}(1 - y_i f(\mathbf{x}_i))_+ + \gamma_1\sum_{i=1}^{\ell}\|\mathbf{w}_i\|_2^2 \\ & + \gamma_2 R(\{b_i,\mathbf{w}_i\}_{i\in I_+}) + \gamma_3 R(\{b_i,\mathbf{w}_i\}_{i\in I_-}),\end{aligned}$$

where I_+ and I_- are the index set for positive and negative examples, respectively. The separation of positive and negative examples reflects our intension to construct a weight graph for each class [3], since we do not expect that a positive example is among the neighbors for a negative example or vice versa. The function values for the positive and negative examples would tend to be far apart, which is clearly an advantage. A more detailed study of the entirely supervised learning case is left for future work.

8 Conclusion

In this paper, we have proposed a new regularization method called tangent space intrinsic manifold regularization, which favors linear functions on the manifold. Local PCA is involved as an important step to estimate tangent spaces and compute the connections between adjacent tangent spaces. Simultaneously, we proposed two new methods for semi-supervised classification with tangent space intrinsic manifold regularization. Experimental results on multiple datasets including comparisons with state-of-the-art algorithms have shown the effectiveness of the proposed methods.

Future work directions include analyzing the generalization error of TiSVMs and TiTSVMs, and applying the tangent space intrinsic manifold regularization to semi-supervised regression and supervised learning tasks.

References

[1] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-supervised Learning*. Cambridge, MA: The MIT Press, 2006.

[2] X. Zhu, ”Semi-supervised learning literature survey,” Department of Computer Sciences, University of Wisconsin-Madison, Madison, WI, Technical Report 1530, Jul. 2008.

[3] M. Belkin, P. Niyogi, and V. Sindhwani, ”Manifold regularization: A geometric framework for learning from labeled and unlabeled examples,” *Journal of Machine Learning Research*, vol. 7, no. 11, pp. 2399–2434, 2006.

[4] R. Johnson and T. Zhang, ”On the effectiveness of Laplacian normalization for graph semi-supervised learning,” *Journal of Machine Learning Research*, vol. 8, no. 7, pp. 1489–1517, 2007.

[5] Z. Qi, Y. Tian, and Y. Shi, ”Laplacian twin support vector machine for semi-supervised classification,” *Neural Networks*, vol. 35, no. 11, pp. 46–53, 2012.

[6] A. Blum and T. Mitchell, ”Combining labeled and unlabeled data with co-training,” in *Proceedings of the 11th Annual Conference on Computational Learning Theory*, Madison, WI, Jul. 1998, pp. 92–100.

[7] J. Farquhar, D. Hardoon, H. Meng, J. Shawe-Taylor, and S. Szedmak, ”Two view learning: SVM-2K, theory and practice,” *Advances in Neural Information Processing Systems*, vol. 18, no. 1, pp. 355–362, 2006.

[8] V. Sindhwani, P. Niyogi, and M. Belkin, ”A co-regularization approach to semi-supervised learning with multiple views,” in *Proceedings of the Workshop on Learning with Multiple Views with the 22nd ICML*, University of Bonn, Germany, Aug. 2005, pp. 1–6.

[9] V. Sindhwani and D. Rosenberg, ”An RKHS for multi-view learning and manifold co-regularization,” in *Proceedings of the 25th International Conference on Machine Learning*, Helsinki, Finland, Jul. 2008, pp. 976–983.

[10] S. Sun, ”Multi-view Laplacian support vector machines,” *Lecture Notes in Computer Science*, vol. 7121, no. 1, pp. 209–222, 2011.

[11] S. Sun and J. Shawe-Taylor, ”Sparse semi-supervised learning using conjugate functions,” *Journal of Machine Learning Research*, vol. 11, no. 12, pp. 2423–2455, 2010.

[12] X. Xie and S. Sun, ”Multi-view Laplacian twin support vector machines,” *Applied Intelligence*, vol. 41, no. 4, pp. 1059–1068, 2014.

[13] A. N. Tikhonov, ”Regularization of incorrectly posed problems,” *Soviet Mathematics Doklady*, vol. 4, no. 6, pp. 1624–1627, 1963.

[14] T. Evgeniou, M. Pontil, and T. Poggio, ”Regularization networks and support vector machines,” *Advances in Computational Mathematics*, vol. 13, no. 1, pp. 1–50, 2000.

[15] C. Hou, F. Nie, F. Wang, C. Zhang, and Y. Wu, ”Semisupervised learning using negative labels,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 22, pp. 420–432, Mar. 2011.

[16] V. N. Vapnik, *Statistical Learning Theory*. New York, NY: Wiley, 1998.

[17] R. Jayadeva, S. Khemchandani, and Chandra, ”Twin support vector machines for pattern classification,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 74, pp. 905–910, May 2007.

[18] R. Tibshirani, ”Regression shrinkage and selection via the lasso,” *Journal of the Royal Statistical Society: Series B Methodological*, vol. 58, no. 1, pp. 267-288, 1996.

[19] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. Cambridge, England: Cambridge University Press, 2004.

[20] D. Rosenberg, *Semi-supervised Learning with Multiple Views*. Ph.D. Thesis, Department of Statistics, University of California, Berkeley, 2008.

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

[22] J. Tenenbaum, V. de Silva, and J. Langford, ”A global geometric framework for nonlinear dimensionality reduction,” *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

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

[24] S. Sun, ”Tangent space intrinsic manifold regularization for data representation,” in *Proceedings of the 1st IEEE China Summit and International Conference on Signal and Information Processing*, Beijing, China, Jul. 2013, pp. 179–183.

[25] I. T. Jolliffe, *Principal Component Analysis*. New York, NY: Springer-Verlag, 1986.

[26] B. Schölkopf and A. Smola, *Learning with Kernels*. Cambridge, MA: The MIT Press, 2001.

[27] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, England: Cambridge University Press, 2004.

[28] J. Shawe-Taylor and S. Sun, ”A review of optimization methodologies in support vector machines,” *Neurocomputing*, vol. 74, no. 17, pp. 3609–3618, 2011.

[29] B. Lin, C. Zhang, and X. He, ”Semi-supervised regression via parallel field regularization,” *Advances in Neural Information Processing Systems*, vol. 24, no. 1, pp. 433–441, 2012.

[30] B. Schölkopf, C. Burges, and V. Vapnik, ”Extracting support data for a given task,” in *Proceedings of the 1st International Conference on Knowledge Discovery and Data Mining*, Montreal, Quebec, Aug. 1995, pp. 252–257.

[31] M. Alvira and R. Rifkin, ”An empirical comparison of SNoW and SVMs for face detection,” Center for Biological and Computational Learning, Massachussetts Institute of Technology, Cambridge, MA, Technical Report 193, Jan. 2001.

[32] S. Sun, ”Ensembles of feature subspaces for object detection,” *Lecture Notes in Computer Science*, vol. 5552, no. 1, pp. 996–1004, 2009.

[33] K. Kim, F. Steinke, and M. Hein, ”Semi-supervised regression using Hessian energy with an application to semi-supervised dimensionality reduction,” *Advances in Neural Information Processing Systems*, vol. 22, no. 1, pp. 979–987, 2010.

Shiliang Sun is a professor at the Department of Computer Science and Technology and the head of the Pattern Recognition and Machine Learning Research Group, East China Normal University. He received the Ph.D. degree from Tsinghua University, Beijing, China, in 2007.

His research interests include kernel methods, learning theory, approximate inference, sequential modeling and their applications, etc.

Xijiong Xie is a Ph.D. student in the Pattern Recognition and Machine Learning Research Group, Department of Computer Science and Technology, East China Normal University.

His research interests include machine learning, pattern recognition, etc.