Locality Preserving Semi-supervised Support Vector Machine

Tongguang Ni1,2, Xiaqing Gu1,2, Shitong Wang2, Pengjiang Qian2,3,4, Raymond F. Muzic Jr3,4.

1. School of Information Science and Engineering, Changzhou University, Changzhou 213164, China
2. School of Digital Media, Jiangnan University, Wuxi 214122, China
3. Case Center for Imaging Research, Case Western Reserve University, Cleveland, Ohio, USA 44106
4. Department of Radiology, University Hospitals Case Medical Center, Case Western Reserve University, Cleveland, Ohio, USA 44106

Abstract: Manifold regularization, which learns from a limited number of labeled samples and a large number of unlabeled samples, is a powerful semi-supervised classifier with a solid theoretical foundation. However, manifold regularization has the tendency to misclassify data near the boundaries of different classes during the classification process. In this paper, we propose a novel classification method called locality preserving semi-supervised support vector machine (LPSSVM) with an extended manifold regularization framework based on within-class locality preserving scatter. LPSSVM is good at exploring the underlying discriminative information as well as the local geometry of the samples as much as possible rather than merely relying on the smoothness information regarding manifold regularization. Meanwhile, benefiting from the geodesic distance metric, LPSSVM can more effectively reflect the true local geometry of data instances in the manifold space, which further strengthens its accuracy in reality. The extensive comparisons with respect to LPSSVM and several state-of-the-art approaches were carried out on both artificial and real-world data sets. These experimental studies demonstrate the advantages as well as the superiority of our proposed method.

Keywords: Manifold regularization; Semi-supervised learning; Support vector machine; Within-class locality preserving scatter

1 Introduction

Support vector machine (SVM) proposed by Vapnik et al. [1-3] is a novel approach for solving pattern recognition problems. Instead of employing empirical risk minimization, SVM implements the structural risk minimization principle that minimizes the upper bound of the generalization error. SVM maps the sample points into a high-dimensional feature space to seek for the optimal separating hyperplane through maximizing the margin between two classes. SVM works with a common assumption: the training data are enough to establish a good model for classification [4]. In reality, however, labeled samples are expensive to obtain in terms of both monetary cost and labeling time in many applications. Whereas, a large amount of unlabeled samples is comparatively cheaper to collect in many domains, such as computer vision, information retrieval, natural language processing and speech recognition.

To overcome this drawback, semi-supervised learning [5-9] has been proposed to exploit the prior knowledge from plenty of unlabeled samples to improve classification performance, such as transductive support vector machine (TSVM) [2], semi-supervised SVM (S3VM) [10], concave convex procedure-TSVM [11], and Laplacian embedded support vector regression (LapESVR) [12]. Similar to the conventional SVM, these existing semi-supervised SVMS dealt with a subset of data points (support vectors) which are close to the decision boundary, and they didn’t consider the underlying geometric structure as well as the discriminant information.

In recent years many manifold learning methods have been proposed, which take advantage of the distribution information of data points. The intrinsic manifold of the data is assumed to embed in an ambient space, such as Fisher linear discriminant analysis (FLDA) [13, 14] and Locality preserving projections (LPP) [15, 16]. FLDA tries to find a linear transformation which maximizes the between-class scatter and minimizes the within-class scatter for separating one class from the others. LPP aims to preserve the local manifold structure of the samples space with modeling manifold structure by a nearest-neighbor graph. Recently, a family of methods called manifold regularization [17, 18] was devised by incorporating the underlying data distribution into the traditional regularization, and many SVMS can be interpreted. The research regarding manifold regularization and SVM has aroused an increasing amount of interest in recent years. For instance, the Laplacian support vector machine (LapSVM) [17], as the most famous representative, was proposed by combining manifold regularization and LPP, attempts to exploit the geometry of the marginal distribution of the samples by penalizing the mapping along the manifold regularization, which results in a significant performance promotion based on semi-supervised linear discriminant analysis.

Manifold regularization is essentially derived from the regression or multivariate functional fitting problems, and it constructs the regularization term primarily in view of the function smoothness. However, in classification, the similar inputs near the discriminant boundaries are more likely to belong to different classes, which imply that just the smoothness constraint may be insufficient for differentiating all classes. In particular, a classifier may not be so smooth
Motivated by this challenge, in this paper, a novel SVM classifier called locality preserving semi-supervised support vector machine (LPSSVM) is proposed. Its steps as well as its distinct merits can be concluded as follows. First, a within-class adjacency graph is built, which can best reflect the geometry structure of the data manifold. Second, the within-class locality preserving scatter matrix is defined, and it is integrated into the manifold regularization framework as a new regularization term. Finally, the optimization problem of LPSSVM is formulated based on the SVM learning framework. Meanwhile, the LPSSVM adopts geodesic distance metric to measure the distance between data in the manifold space, which can reflect the true geometry of the manifold. We have conducted extensive experiments on both artificial and real world data sets, and the experimental results demonstrate that LPSSVM achieves better prediction performance compared with the standard SVM, the minimum class locality preserving variance support vector machine (MCLPV-SVM) [20], the semi-supervised discriminant analysis (SDA) [21] and the LapSVM.

The rest of this paper is organized as follows. In Section 2, we briefly review the manifold regularization and LapSVM. Our proposed LPSSVM in the linear and nonlinear case are presented in Section 3, respectively. The experiments are described in Section 4, and some concluding remarks are given in Section 5.

2 Manifold regularization

The manifold regularization framework extends the classical framework of regularization in reproducing kernel Hilbert spaces (RKHS) to exploit the geometry of the marginal distribution with incorporation of both labeled and unlabeled samples [17, 18]. Let \( S = \{x_i \in \mathbb{R}^d, i = 1, \ldots, l + u\} \) denote the set of training samples, consisting of \( l \) labeled samples and \( u \) unlabeled samples. Suppose \( L = \{(y_i, y_j)\}_{i=1}^{l} \) is the set of labeled samples drawn according to the joint distribution \( P_{xy} \), and \( U = \{(x_i)_{i=1}^{u}\} \) is the set of unlabeled samples drawn according to the marginal distribution \( P_x \) of \( P_{xy} \). Since \( P_x \) is unknown in most applications, the manifold regularization framework must get empirical estimates of \( P_x \) using a large number of unlabeled samples and then constrain the conditional distribution \( P(y | x) \) with a few labeled samples, so the relationship between them must be assumed. Manifold regularization [17] assumes that the probability distribution of data has the geometric structure of a Riemannian manifold \( M \). The conditional distributions \( P(y | x) \) and \( P(y | x_i) \) are similar or same if the two data points \( x_1, x_2 \) are close in the intrinsic geometry of \( P_x \) (i.e., with respect to geodesic distances on \( M \)). An intrinsic regularizer \( \|f\|_M \), which reflect the locality information of data, is empirically estimated from the point cloud of labeled and unlabeled samples using the graph Laplacian. In this regard, the graph is an approximation of the manifold \( M \), where a node \( x \) in the graph is a point in \( M \) and the weight \( W_{xy} \) on an edge connecting two points \( x_i \) and \( x_j \) is the adjacency of the nodes. In particular, choosing exponential weights for the adjacency matrix leads to convergence of the graph Laplacian to the Laplace-Beltrami operator on the manifold [22, 23]. Therefore, we consider

\[
\|f\|_M^2 = \sum_{i=1}^{l} \sum_{j=1}^{u} W_{ij} (f(x_i) - f(x_j))^2 = \sum_{ij} W_{ij} (f_j - f_i)^2 = f^T L f ,
\]

where \( W_{ij} \) is the weight measuring the similarity between data points \( x_i \) and \( x_j \), \( L = D - W \) is the graph Laplacian matrix, and \( D \) is a diagonal matrix with its entries \( D_{ii} = \sum_{i=1}^{l} W_{ij} \).

Given a Mercer kernel \( K \) and its induced RKHS \( H_K \), the objective function of manifold regularization is

\[
\min_{f \in H_K} \frac{1}{l} \sum_{i=1}^{l} V(x_i, y_i, f(x_i)) + \gamma_r \|f\|^2_2 + \gamma_r \|f\|^2_2 ,
\]

where \( V(\cdot) \) is a certain loss function, the regularization term \( \|f\|^2_2 \) is employed to control the complexity of the classifier to avoid overfitting, \( f(x) \) is the decision function, \( \gamma_r \) and \( \gamma_r \) are two regularization coefficients.

If \( V(\cdot) \) in Eq.(2) is the hinge loss function \( \max[0, 1 - y_i f(x_i)] \) for SVM, the manifold regularization is extended to the Laplacian support vector machines (LapSVM) [17], thus, the primal problem in Eq.(2) becomes

\[
\min_{a \in \mathbb{R}^l, z \in \mathbb{R}^l} \frac{1}{l} \sum_{i=1}^{l} z_i + \gamma_r a^T K a + \gamma_r \frac{1}{u+l} a^T K L K a \\
\text{s.t.} \quad y_i (\sum_{j=1}^{l} a_j K(x_i, x_j) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, l ,
\]

where \( K \) is a kernel matrix with entry \( K_{ij} = K(x_i, x_j) = \phi(x_i)^T \phi(x_j) \).

By the Representer Theorem, the solution to the problem above is given by
\[
f(x) = \sum_{i=1}^{l_u} \alpha_i K(x, x_i) + b.
\] (4)

3 Locality preserving semi-supervised support vector machine (LPSSVM)

The motivation for LPSSVM arises from the following considerations.

In our view, manifold regularization still has two flaws in theory. First, manifold regularization constructs the regularization term by focusing mainly on the smoothness of the function. If two samples \(x_i, x_j \in X\) are close in the intrinsic geometry of \(P_x\), \(P(y \mid x_i)\) and \(P(y \mid x_j)\) are same or similar. Obviously, this assumption violates the fact that two neighboring samples lying across but near the boundaries belong to different classes. Second, although manifold regularization enhances similarity relationships between nearby samples, it ignores the structure information in the manifold separation of different classes. To address these issues, in this paper, we propose a novel method LPSSVM, in which the local geometrical structure and the intrinsic manifold of within-class data are considered.

3.1 Within-class neighborhood preserving scatter matrix

For a set \(X = \{x_i\} (i = 1, 2, ..., l + u)\), we build a nearest neighbor graph \(G\) over the \((l+u)\) samples to model intrinsic geometrical structure of data manifold. For each example \(x_i\), we seek for its \(k\) nearest neighbors set \(KNN(i)\) and put an edge between \(x_i\) and its neighbors. Thus, the corresponding weight matrix of graph \(G\) can be defined as

\[
W_{ij} = \begin{cases} 1 & \text{if } i \in KNN(j) \text{ or } j \in KNN(i) \\ 0 & \text{otherwise} \end{cases},
\] (5)

The Laplacian matrix \(L = D - W\), where \(D\) is a diagonal matrix whose entries are column (or row) sum of \(W\), i.e. \(D = \sum_{i} W_{ii}\).

For effectively characterizing the local geometry of the data manifold where the samples lie on, according to [24, 25], we further devise the within-class graph \(G_w\), with weight matrix of graph \(G_w\) as

\[
W_{w,ij} = \begin{cases} \exp \left( -\frac{d^2(x_i, x_j)}{t} \right) & \text{if } i \in KNN(j) \text{ or } j \in KNN(i) \text{ and } y_i = y_j \\ 0 & \text{otherwise} \end{cases},
\] (6-1)

\[
d(x_i, x_j) = \exp \left( -\frac{d^2(x_i, x_j)}{t} \right),
\] (6-2)

where \(\exp(-d^2(x_i, x_j)/t)\) is called the heart kernel function, and \(t\) is a constant. It is noted that \(d(x_i, x_j)\) is the geodesic distance [26] between point \(x_i\) and point \(x_j\). \(d_s(x_i, x_j)\) denotes the edge lengths in graph \(G_w\). Initialize \(d(x_i, x_j) = d_s(x_i, x_j)\), if point \(x_i\) and point \(x_j\) are linked by an edge; otherwise, \(d(x_i, x_j) = \infty\). Then, \(d(x_i, x_j)\) can be computed by \(\min\{d_s(x_i, x_j), d_s(x_i, x_j) + d_s(x_j, x_j)\} (0 < t \leq k)\).

Moreover, to characterize the within-class compactness in output space, we define the within-class locality preserving scatter as

\[
\tilde{S}_w = \sum_{i=1}^{l_u} \sum_{j=1}^{l+u} \left\| f(x_i) - f(x_j) \right\|^2 W_{w,ij} = 2 f^T (D_w - W_w) f = 2 f^T L_w f,
\] (7)

where \(D_w\) is the diagonal matrix with entries equaling column (or row) sum of \(W_w\), i.e. \(D_w = \sum_{j=1}^{l+u} W_{w,ij}\), and \(L_w = D_w - W_w\) is the within-class Laplacian matrix of \(G_w\).

The reason why we use geodesic distance instead of Euclidean distance is that the geometry of the high-dimension input space may be highly folded, twisted, or curved, so that the Euclidean distance may not accurately reflect their intrinsic similarity; meanwhile, the geodesic distance can represent the shortest path in the high-dimension space [26]. In a word, the geodesic distance can more reliably reflect the true high-dimension geometry of the manifold.

3.2 LPSSVM in the linear case

Suppose \(l\) labeled samples \(\{(x_i, y_i)\}_{i=1}^{l} \in X\) and \(u\) unlabeled samples \(\{(x_j)\}_{j=1}^{u} \in X\) are given. By incorporating both the manifold regularization framework and the knowledge of underlying data distribution, we design the objective function of manifold regularization

\[
\min_{f \in \mathbb{R}^d} \frac{1}{l} \sum_{i=1}^{l} V(x_i, y_i, f(x_i)) + \gamma_l \left\| f \right\|^2 + \gamma_i \left\| f \right\|_1^\alpha + \frac{\gamma_d}{2} \tilde{S}_w,
\] (8)

where \(\sum_{i=1}^{l} V(x_i, y_i, f(x_i))\) is the loss function of labeled samples, which is defined as the hinge loss.
(1 − yf(x)) = \max(0, 1 − yf(x)) and the labels y, \in \{-1, +1\}. The ambient regularizer \|f\|_2^2 controls the complexity of the classifier to avoid overfitting. The manifold regularizer \|f\|_2^2 is similar to Eq. (1), which makes P(y | x) vary smoothly by means of incurring a heavy penalty for those data points which are close in the input space while lie far way in the output space.

A reasonable classification function should be smooth within each class, as well as maximize the margins between the samples of different classes simultaneously. Thus, In this regard, the term \(\frac{\gamma_f}{2} \tilde{S}_w^2\) in Eq. (8) plays the role of correcting the assumption of “always smooth”.

As a result, our objective can be reformulated as

\[
\min_{f, b / h} \frac{1}{l} \sum_{i=1}^{l} (1 - y_i f(x_i)) + \gamma_A \|f\|_2^2 + \gamma_f f^T L f + \gamma_D f^T L_d f ,
\]

Here we simply assume that the classifier has a linear form

\[
f(x) = w^T x + b .
\]

Based on linear manifold regularization framework \([17]\) and the SVM learning framework, we can obtain the primal of the linear LPSSVM

\[
\min_{w, b / \xi_i / r / r_l} \frac{1}{l} \sum_{i=1}^{l} (1 - y_i (w^T x_i + b)) + \frac{\gamma_A}{2} w^T w + \frac{\gamma_f}{2} w^T X L X^T w + \frac{\gamma_D}{2} w^T X_l X^T w ,
\]

where \(\gamma_A, \gamma_f\) and \(\gamma_D\) are three parameters balancing different terms.

Then, the linear LPSSVM can be easily achieved by solving the following optimization problem:

\[
\min_{w, b / \xi_i / r / r_l} \frac{1}{l} \sum_{i=1}^{l} \xi_i + \frac{\gamma_A}{2} w^T w + \frac{\gamma_f}{2} w^T X L X^T w + \frac{\gamma_D}{2} w^T X_l X^T w \\
\text{s.t.} \quad y_i \sum_{i=1}^{l} (w, x_i) + b \geq 1 - \xi_i, \xi_i \geq 0, \quad i = 1, ..., l .
\]

By introducing the Lagrange multipliers \((\beta = (\beta_1, \beta_2, ..., \beta_l), \gamma = (\gamma_1, \gamma_2, ..., \gamma_l))\) into the inequality constrains of Eq.(12), the corresponding Lagrangian L is

\[
L(w, b, \xi, \beta, r) = \frac{1}{l} \sum_{i=1}^{l} \xi_i + \frac{\gamma_A}{2} w^T w + \frac{\gamma_f}{2} w^T X L X^T w + r^T (\sum_{i=1}^{l} \beta_i y_i (w^T x_i + b) - 1 + \xi_i - \sum_{i=1}^{l} \gamma_i) .
\]

By setting the derivatives of Eq. (13) with respect to all the variables to zeros, we have

\[
\frac{\partial L}{\partial \beta_i} = 0 \implies \frac{1}{l} \beta_i - \gamma_i = 0 \implies 0 \leq \beta_i \leq \frac{1}{l} ,
\]

\[
\frac{\partial L}{\partial \gamma_i} = 0 \implies \sum_{i=1}^{l} \beta_i y_i = 0 ,
\]

\[
\frac{\partial L}{\partial \gamma_i} = 0 \implies (\gamma_A + \gamma_f X L X^T + \gamma_D X_l X^T) w - \sum_{i=1}^{l} \beta_i y_i x_i = 0 .
\]

Substituting these equations back into Eq. (13), we can arrive at the dual of Eq. (12) as

\[
\min_{\beta / \gamma} \frac{1}{2} \beta^T H \beta - \sum_{i=1}^{l} \gamma_i
\]

\[
\text{s.t.} \quad \beta^T Y = 0, \quad \beta_i \in [0, \frac{1}{l}] ,
\]

where \(Y = \text{diag}(y_1, y_2, ..., y_l), \quad H = (h_{ij}), \quad h_{ij} = y_i y_j x_i^T (r_j I + r_j X L X^T + r_l X_l X^T)^{-1} x_j \) with \(I\) being \(l \times l\) identity matrix.

Eq.(17) is a typical convex quadratic programming (QP) problem which is easy to be numerically solved. Suppose \(\beta^* = (\beta_1^*, ..., \beta_l^*)\) is the optimum solution of Eq. (17), then the optimal weight vector \(w\) and \(b\) can be represented as

\[
w^* = \sum_{i=1}^{l} \beta_i^* y_i (r_j I + r_l X L X^T + r_l X_l X^T)^{-1} x_i ,
\]

\[
b^* = \sum_{i=1}^{l} (y_i - \sum_{i=1}^{l} \beta_i^* y_i x_i^T (r_j I + r_j X L X^T + r_l X_l X^T)^{-1} x_j) .
\]

As a result, the corresponding decision function of the linear LPSSVM should be

\[
f(x) = w^T x + b^* = \sum_{i=1}^{l} \beta_i^* y_i (x_i^T (r_j I + r_j X L X^T + r_l X_l X^T)^{-1} x) + b^* .
\]

It is worth noting that to deal with the problem of small data size, \(r_j I + r_l X L X^T + r_l X_l X^T\) is regularized by
combining $r_1XL^T + r_2XL_n^T$ with $r_1I$ before any inversion computing. Hence, $r_1I + r_2XL^T + r_2XL_n^T$ is always nonsingular, and its inverse exists.

### 3.3 LPSSVM in the nonlinear case

Now we propose the nonlinear LPSSVM. In the light of the properties of RKHS, the classical Representer Theorem states that the solution to the above minimization problem exists in RKHS and $w$ can be written as

$$w = \sum_{i=1}^{l} \alpha_i \phi(x_i),$$

and then

$$f(x) = \sum_{i=1}^{l} \alpha_i K(x, x_i),$$

(21)

Based on the SVM learning framework, we introduce an unregularized bias $b$ to Eq. (21). Correspondingly, the LPSSVM model can be described as the following optimization problem

$$\min_{\alpha \in \mathbb{R}^l, \gamma \geq 0} \frac{1}{2} \sum_{i=1}^{l} \xi_i^2 + \gamma \alpha^T Ka + \frac{\gamma}{2} \alpha^T KLK \alpha$$

s.t. $y_i(\sum_{j \neq i} \alpha_j K(x_j, x_i) + b) \geq 1 - \xi_i, \xi_i \geq 0, \quad i = 1, \ldots, l.$

By means of Lagrangian optimization, we can attain the dual form of Eq. (22)

$$\min_{\beta \in \mathbb{R}^l} \frac{1}{2} \beta^T Q \beta - \sum_{i=1}^{l} \beta_i$$

s.t. $\beta^T Y = 0, \quad \beta_i \in [0, 1].$

where $Q = YJK(\gamma_A I + \gamma_I LK + \gamma_D L_n K)^{-1} J^T Y, \quad Y = \text{diag}(y_1, y_2, \ldots, y_l), \quad J = [I, 0]_{l \times l}$ with $I$ being $l \times l$ identify matrix.

After solving for $\beta^*$, we can derive subsequently the optima of $\alpha^*$ and $b^*$ as

$$\alpha^* = (\gamma_A I + \gamma_I LK + \gamma_D L_n K)^{-1} J^T Y \beta^*,$$

(24)

$$b^* = \frac{1}{l} \sum_{i=1}^{l} (y_i - \sum_{j=1}^{l} \alpha_j^* K(x_j, x_i)),$$

(25)

Moreover, the final decision function can be written as

$$f^*(x) = \sum_{i=1}^{l} \alpha_i^* K(x, x_i) + b^*.$$  

(26)

### 3.4 The algorithm of LPSSVM in the linear and nonlinear case

Based on the above, we can state the proposed LPSSVM as follows.

<table>
<thead>
<tr>
<th>Learning algorithm for LPSSVM</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> $l$ labeled samples ${ (x_i, y_i) }<em>{i=1}^{l}$, $u$ unlabeled samples ${ x_j }</em>{j=l+1}^{l+u}$.</td>
</tr>
<tr>
<td><strong>Output:</strong> Decision function $f(x)$.</td>
</tr>
<tr>
<td><strong>Step 1</strong> Construct a KNN adjacent graph with $(l + u)$ nodes and compute the edge weights matrix $W$.</td>
</tr>
<tr>
<td><strong>Step 2</strong> Compute graph Laplacian matrix $L = D - W$.</td>
</tr>
<tr>
<td><strong>Step 3</strong> Construct the within-class graph $G_w$ and compute the edge weights matrix $W_w$.</td>
</tr>
<tr>
<td><strong>Step 4</strong> Compute within-class graph Laplacian matrix $L_w = D_w - W_w$.</td>
</tr>
<tr>
<td>In linear case, Compute $\beta^<em>$ by solving Eq.(17) with a QP Solver, Compute $\alpha^</em>$ using Eq.(18), compute $b^*$ using Eq.(19);</td>
</tr>
<tr>
<td><strong>Step 5</strong> In nonlinear case, choose a kernel function $K(x,y)$. Compute the Gram matrix $K_{ij} = K(x_i, x_j)$, Compute $\beta^<em>$ by solving Eq.(23) with a QP Solver, Compute $\alpha^</em>$ using Eq.(24), compute $b^*$ using Eq.(25).</td>
</tr>
<tr>
<td><strong>Step 6</strong> In linear case, using decision function Eq.(20) with samples $x_i$, in nonlinear case, using decision function Eq.(26) with samples $x_i$, and output the final class labels.</td>
</tr>
</tbody>
</table>
4 Experiments and discussions

To evaluate the performance of our proposed methods, in this section we systematically compare LPSSVM with several state-of-the-art classification methods on different data sets, including the standard SVM, MCLPV_SVM, SDA and LapSVM. First, we compared MCLPV_SVM, LapSVM with LPSSVM on a series of two-dimensional two-moon data sets with different complexities and different labeled points. Second, all the involved methods were implemented on twenty artificial and UCI real-word data sets [27, 28], among them, WindowsMac and AutosHockey are generated from 20NewsGroups, OrgsPlaces, PlacesPeople and OrgsPeople are generated form Reuters-21578. Finally, we carried out all the methods for face recognition with the Yale face database [29] and the CMU PIE face database [30]. We implement SVM using Libsvm [31] and the other methods using MATLAB. All the experiments were performed with Matlab (R2010a) in a PC with configuration as follows: CPU 2.99 GHz, 8.0 G RAM and Microsoft Windows 7.

4.1 Experimental settings and parameter selection

Before describing the results of our experiments, we should first explain the parameter setting of each method in our study. In general, the commonest grid search and four-fold cross-validation strategies were adopted to obtain the best parameter values in our study. More specifically, the ranges of the primary parameters in each approach were set as follows. The regularization parameter \( C \) of the standard SVM and MCLPV_SVM varied within the set \{0.001, 0.01, 0.1, 1, 10, 100\}. The neighborhood parameter \( k \) in MCLPV_SVM, SDA, LapSVM and LPSSVM, was selected from the set \{1, 3, 5, 7, 9, 11\}, and the heat kernel parameter \( t \) took values from the set \{0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0\}. The optimal parameters \( \gamma_A, \gamma_D \) of LapSVM and LPSSVM were limited in the set \{10\(^{-6}\), 10\(^{-4}\), 10\(^{-2}\), 1, 10, 100\}. The parameter \( \alpha \) in SDA was chosen in the set \{2\(^{-7}\), 2\(^{-6}\), 2\(^{-5}\), 2\(^{-4}\), 1, 2\(^{1}\), 2\(^{2}\), 2\(^{3}\), 2\(^{4}\)\}. Moreover, the parameter \( \gamma_D \) in LPSSVM was tried in the interval \[0, 1\]. For the nonlinear case of LPSSVM, the Gaussian RBF kernel was employed in our research, i.e., \( \exp(-((v-u)^T(v-u)/\sigma)) \), where \( \sigma \) is the spread of Gaussian kernel, and \( \sigma \) was searched in \([\tau^2/16, \tau^2/8, \tau^2/4, \tau^2/2, \tau^2, 2\tau^2, 4\tau^2, 8\tau^2, 16\tau^2]\), where \( \tau^2 \) is the mean norm of the training data.

4.2 Experiments of two-moon data sets

In this subsection, we construct a series of two-moon data sets to justify the basic rational of the proposed method LPSSVM. The two-moon data sets demonstrate the superiority of semi-supervised discriminative regularization method. We compare MCLPV_SVM and LapSVM with LPSSVM.

First, three two-moon data sets with different complexities are discussed in Figure 1. Each data set contains 150 labeled samples indicated with red color and blue color respectively in each class. The number of the \( k \) nearest neighbors is fixed to \( 5 \).

Second, three two-moon data sets with different number of labeled samples are discussed in Figure 2. Similarly, each data set contains 150 samples in each class and 16, 4, 1 labeled sample respectively in each class. The labeled samples are indicated with different colors (red and blue). White points are unlabeled.

From Figure 1, it can be seen that: (1) As a traditional manifold regularization method, the decision hyperplane derived from LapSVM is smooth along all of the two-moon data sets ((A.2), (B.2), (C.2)). However, as the two moons move close to each other, LapSVM tends to misclassify the points near boundaries. Since LapSVM firmly believes neighboring points are supposed to be in the same class, it is clearly seen that LapSVM is over-smooth in the data set (C.2). (2) When the two moons are far apart from each other, the boundaries of LPSSVM ((A.3)) are adequately smooth as well as LapSVM. But when the two moons are close to each other, the decision hyperplanes of MCLPV_SVM and LPSSVM become not smooth ((C.1), (C.3)). It can be seen that MCLPV_SVM and LPSSVM can make a better decision about the data near boundaries in the data set ((C.1), (C.3)), which clearly justifies the intuition that, the underlying discriminative information in the within-class is vital for classification. Meanwhile, the smoothness assumption in the traditional regularization method is too general for classification. When the two moons include overlapping points ((C.2), (C.3)), the effectiveness of the matrix \( W_c \) is not obvious, based on the manifold information, almost only the SVM mechanism playing roles in both LapSVM and LPSSVM, so the performance of LPSSVM is only slightly better than that of LapSVM.

From Figure 2, it can be seen that: (1) The performance of LPSSVM and LapSVM depends heavily on not only the number of labeled samples but also on the distribution of labeled samples, but MCLPV_SVM only utilizes the labeled samples and leaves unlabeled samples unused. Therefore, MCLPV_SVM fails to discover the optimal boundaries when label information is not sufficient ((B.1), (C.1)). (2) Since LPSSVM considers the local manifold structure of the samples, the decision hyperplanes of LPSSVM accord with more the geometry of the samples, and it always obtains good performance in all cases. It can be also seen that since only two labeled points are available in the data sets ((C.2), (C.3)), consequently, the \( W_c \) is extremely sparse. Except for SVM, the classification performance of LPSSVM was solely dependent on manifold information during the entire classification, so LPSSVM and LapSVM obtain almost the same classification results.
Figure 1. The decision hyperplanes in the three two-moon data sets with different complexities.
(MCLPV_SVM (A.1), (B.1), (C.1)), LapSVM ((A.2), (B.2), (C.2)) and LPSSVM ((A.3), (B.3), (C.3))

Figure 2. The decision hyperplanes in the three two-moon data sets with different labeled points.
(MCLPV_SVM (A.1), (B.1), (C.1)), LapSVM ((A.2), (B.2), (C.2)) and LPSSVM ((A.3), (B.3), (C.3))

4.3 Experiments of different artificial and real-word data sets
To further investigate the classification performance of LPSSVM, we compare it with SVM, MCLPV_SVM, SDA and LapSVM in several artificial and real-world data sets. All the data sets are binary-class data sets. Further details of these data sets are provided in Table 1. All data have been normalized before experiment. For each data set, we divide the samples into two non-overlapping training set and testing set, and each set contains almost half of the samples in each class respectively. This process is repeated ten times to generate ten independent runs for each data set, and then
the average results are collected and reported. First, we report the obtained experimental results of the benchmarking methods on these 20 data sets with the fixed numbers of labeled and unlabeled samples in the linear and nonlinear case respectively, and results are shown in Table 2 and Table 3. Then, we report the change tendency of the performance of all the methods by running them on the adopted data sets with varying number of labeled samples and unlabeled samples. In order to save space, we only report the obtained experimental results on the data sets of G241c, G241n, Australia, Optigidits, Shuttle and WindowsMac.

<table>
<thead>
<tr>
<th>Data sets</th>
<th>Dimension</th>
<th>Training set labelled</th>
<th>Training set unlabelled</th>
<th>Testing set</th>
</tr>
</thead>
<tbody>
<tr>
<td>g241c</td>
<td>241</td>
<td>100</td>
<td>1000</td>
<td>400</td>
</tr>
<tr>
<td>g241n</td>
<td>241</td>
<td>100</td>
<td>1000</td>
<td>400</td>
</tr>
<tr>
<td>Australia</td>
<td>14</td>
<td>60</td>
<td>400</td>
<td>460</td>
</tr>
<tr>
<td>German_org</td>
<td>24</td>
<td>30</td>
<td>250</td>
<td>666</td>
</tr>
<tr>
<td>Heart</td>
<td>13</td>
<td>20</td>
<td>160</td>
<td>90</td>
</tr>
<tr>
<td>Ionoshere</td>
<td>33</td>
<td>20</td>
<td>210</td>
<td>117</td>
</tr>
<tr>
<td>Liver_disorder</td>
<td>6</td>
<td>30</td>
<td>200</td>
<td>115</td>
</tr>
<tr>
<td>Glass2</td>
<td>9</td>
<td>10</td>
<td>95</td>
<td>55</td>
</tr>
<tr>
<td>Sonar</td>
<td>60</td>
<td>15</td>
<td>120</td>
<td>70</td>
</tr>
<tr>
<td>Monk2</td>
<td>6</td>
<td>25</td>
<td>140</td>
<td>432</td>
</tr>
<tr>
<td>Breast</td>
<td>9</td>
<td>20</td>
<td>80</td>
<td>170</td>
</tr>
<tr>
<td>Hepatitis</td>
<td>19</td>
<td>10</td>
<td>30</td>
<td>115</td>
</tr>
<tr>
<td>Diabetes</td>
<td>8</td>
<td>68</td>
<td>500</td>
<td>100</td>
</tr>
<tr>
<td>Optigidits</td>
<td>42</td>
<td>80</td>
<td>800</td>
<td>163</td>
</tr>
<tr>
<td>Shuttle</td>
<td>9</td>
<td>100</td>
<td>10000</td>
<td>1000</td>
</tr>
<tr>
<td>WindowsMac</td>
<td>7511</td>
<td>160</td>
<td>1600</td>
<td>228</td>
</tr>
<tr>
<td>AutosHockey</td>
<td>6398</td>
<td>160</td>
<td>1600</td>
<td>236</td>
</tr>
<tr>
<td>OrgsPlaces</td>
<td>6889</td>
<td>100</td>
<td>1000</td>
<td>139</td>
</tr>
<tr>
<td>PlacesPeople</td>
<td>6226</td>
<td>80</td>
<td>800</td>
<td>136</td>
</tr>
<tr>
<td>OrgsPeople</td>
<td>6206</td>
<td>80</td>
<td>800</td>
<td>199</td>
</tr>
</tbody>
</table>

Table 2. Classification accuracy (%) and standard deviation compared between SVM, MCLPV_SVM, SDA, LapSVM and LPSSVM on the different data sets in the linear case.

<table>
<thead>
<tr>
<th>Data sets</th>
<th>SVM</th>
<th>MCLPV_SVM</th>
<th>SDA</th>
<th>LapSVM</th>
<th>LPSSVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>g241c</td>
<td>73.12±0.076</td>
<td>76.21±0.073</td>
<td>77.98±0.092</td>
<td>78.01±0.10</td>
<td>80.71±0.09</td>
</tr>
<tr>
<td>g241n</td>
<td>72.72±0.099</td>
<td>75.05±0.09</td>
<td>76.12±0.092</td>
<td>76.08±0.12</td>
<td>78.75±0.08</td>
</tr>
<tr>
<td>Australia</td>
<td>58.33±0.042</td>
<td>59.15±0.037</td>
<td>60.67±0.017</td>
<td>60.98±0.009</td>
<td>61.57±0.017</td>
</tr>
<tr>
<td>German_org</td>
<td>53.17±0.045</td>
<td>53.14±0.031</td>
<td>54.21±0.044</td>
<td>55.87±0.093</td>
<td>56.78±0.022</td>
</tr>
<tr>
<td>Heart</td>
<td>80.19±0.060</td>
<td>80.21±0.042</td>
<td>82.67±0.027</td>
<td>81.78±0.082</td>
<td>82.39±0.043</td>
</tr>
<tr>
<td>Ionoshere</td>
<td>80.09±0.019</td>
<td>80.11±0.071</td>
<td>81.33±0.061</td>
<td>83.56±0.055</td>
<td>84.67±0.018</td>
</tr>
<tr>
<td>Liver_disorder</td>
<td>64.80±0.039</td>
<td>64.98±0.048</td>
<td>65.69±0.070</td>
<td>67.47±0.065</td>
<td>67.99±0.032</td>
</tr>
<tr>
<td>Glass2</td>
<td>62.22±0.082</td>
<td>62.76±0.055</td>
<td>65.34±0.033</td>
<td>65.98±0.028</td>
<td>65.98±0.077</td>
</tr>
<tr>
<td>Sonar</td>
<td>61.88±0.023</td>
<td>63.43±0.027</td>
<td>68.88±0.058</td>
<td>70.65±0.011</td>
<td>71.87±0.016</td>
</tr>
<tr>
<td>Monk2</td>
<td>67.67±0.068</td>
<td>67.53±0.029</td>
<td>67.67±0.008</td>
<td>68.77±0.074</td>
<td>70.89±0.049</td>
</tr>
<tr>
<td>Breast</td>
<td>88.89±0.072</td>
<td>88.98±0.050</td>
<td>93.92±0.021</td>
<td>93.01±0.082</td>
<td>93.91±0.035</td>
</tr>
<tr>
<td>Hepatitis</td>
<td>69.71±0.053</td>
<td>70.77±0.041</td>
<td>70.98±0.089</td>
<td>73.87±0.030</td>
<td>73.90±0.050</td>
</tr>
<tr>
<td>Diabetes</td>
<td>55.67±0.11</td>
<td>57.67±0.10</td>
<td>57.21±0.098</td>
<td>57.87±0.10</td>
<td>58.85±0.11</td>
</tr>
<tr>
<td>Optigidits</td>
<td>87.90±0.14</td>
<td>90.55±0.11</td>
<td>90.12±0.11</td>
<td>90.14±0.12</td>
<td>91.05±0.10</td>
</tr>
<tr>
<td>Shuttle</td>
<td>80.65±0.10</td>
<td>85.11±0.91</td>
<td>86.05±0.10</td>
<td>86.13±0.11</td>
<td>87.53±0.11</td>
</tr>
<tr>
<td>WindowsMac</td>
<td>78.11±1.23</td>
<td>85.22±0.78</td>
<td>85.79±1.08</td>
<td>85.77±1.07</td>
<td>86.31±0.98</td>
</tr>
<tr>
<td>AutosHockey</td>
<td>73.77±1.00</td>
<td>76.39±0.67</td>
<td>78.44±0.89</td>
<td>78.91±1.21</td>
<td>79.56±1.15</td>
</tr>
<tr>
<td>OrgsPlaces</td>
<td>78.56±0.89</td>
<td>80.11±0.64</td>
<td>80.01±0.88</td>
<td>80.81±0.85</td>
<td>81.33±1.11</td>
</tr>
<tr>
<td>PlacesPeople</td>
<td>68.15±0.52</td>
<td>75.99±1.03</td>
<td>75.47±0.93</td>
<td>75.65±0.78</td>
<td>76.60±1.09</td>
</tr>
<tr>
<td>OrgsPeople</td>
<td>66.87±0.29</td>
<td>67.98±1.35</td>
<td>68.11±0.79</td>
<td>68.06±0.25</td>
<td>68.57±0.88</td>
</tr>
</tbody>
</table>
Table 3. Classification accuracy (%) and standard deviation compared between SVM, MCLPV_SVM, SDA, LapSVM and LPSSVM on the different data sets with the RBF kernel.

<table>
<thead>
<tr>
<th>Data sets</th>
<th>SVM</th>
<th>MCLPV_SVM</th>
<th>SDA</th>
<th>LapSVM</th>
<th>LPSSVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>g241c</td>
<td>76.12±0.092</td>
<td>78.69±0.077</td>
<td>80.25±0.12</td>
<td>80.11±0.10</td>
<td>82.12±0.12</td>
</tr>
<tr>
<td>g241n</td>
<td>75.72±0.081</td>
<td>78.21±0.08</td>
<td>79.01±0.088</td>
<td>78.99±0.15</td>
<td>80.35±0.14</td>
</tr>
<tr>
<td>Australia</td>
<td>61.33±0.015</td>
<td>63.15±0.047</td>
<td>65.67±0.049</td>
<td>65.98±0.037</td>
<td>67.00±0.020</td>
</tr>
<tr>
<td>German_org</td>
<td>56.97±0.025</td>
<td>57.14±0.024</td>
<td>62.21±0.039</td>
<td>63.87±0.066</td>
<td>64.65±0.057</td>
</tr>
<tr>
<td>Heart</td>
<td>85.89±0.034</td>
<td>85.21±0.019</td>
<td>86.67±0.054</td>
<td>86.78±0.025</td>
<td>86.78±0.025</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>79.01±0.019</td>
<td>82.76±0.026</td>
<td>83.33±0.027</td>
<td>85.56±0.052</td>
<td>85.99±0.048</td>
</tr>
<tr>
<td>Liver_disorder</td>
<td>69.80±0.083</td>
<td>70.98±0.030</td>
<td>71.69±0.084</td>
<td>71.47±0.017</td>
<td>72.60±0.069</td>
</tr>
<tr>
<td>Glass2</td>
<td>72.22±0.074</td>
<td>72.76±0.056</td>
<td>73.34±0.026</td>
<td>73.76±0.080</td>
<td>73.76±0.009</td>
</tr>
<tr>
<td>Sonar</td>
<td>64.88±0.063</td>
<td>68.43±0.028</td>
<td>73.18±0.038</td>
<td>70.65±0.092</td>
<td>72.96±0.073</td>
</tr>
<tr>
<td>Monk2</td>
<td>70.67±0.026</td>
<td>67.33±0.017</td>
<td>71.67±0.040</td>
<td>71.77±0.034</td>
<td>73.89±0.052</td>
</tr>
<tr>
<td>Breast</td>
<td>89.12±0.055</td>
<td>92.98±0.092</td>
<td>92.23±0.076</td>
<td>94.01±0.016</td>
<td>94.82±0.021</td>
</tr>
<tr>
<td>Hepatitis</td>
<td>73.45±0.010</td>
<td>74.97±0.023</td>
<td>76.11±0.074</td>
<td>78.17±0.031</td>
<td>78.80±0.037</td>
</tr>
<tr>
<td>Diabetes</td>
<td>58.83±0.12</td>
<td>59.41±0.12</td>
<td>59.01±0.108</td>
<td>59.09±0.12</td>
<td>60.75±0.10</td>
</tr>
<tr>
<td>Optdigits</td>
<td>94.90±0.11</td>
<td>98.15±0.10</td>
<td>98.20±0.11</td>
<td>98.23±0.12</td>
<td>99.20±0.099</td>
</tr>
<tr>
<td>Shuttle</td>
<td>85.75±1.09</td>
<td>90.75±0.59</td>
<td>91.17±0.11</td>
<td>91.29±0.12</td>
<td>92.33±0.12</td>
</tr>
<tr>
<td>WindowsMac</td>
<td>80.22±1.23</td>
<td>88.89±1.63</td>
<td>88.44±1.01</td>
<td>88.56±1.01</td>
<td>89.71±0.72</td>
</tr>
<tr>
<td>AutosHockey</td>
<td>76.77±1.00</td>
<td>78.99±0.98</td>
<td>81.34±0.73</td>
<td>82.45±1.88</td>
<td>83.67±1.01</td>
</tr>
<tr>
<td>OrgsPlaces</td>
<td>80.18±0.78</td>
<td>82.33±0.56</td>
<td>81.99±0.98</td>
<td>82.19±0.93</td>
<td>83.22±1.03</td>
</tr>
<tr>
<td>PlacesPeople</td>
<td>70.45±1.23</td>
<td>77.37±1.23</td>
<td>77.51±0.73</td>
<td>77.89±0.81</td>
<td>78.96±1.00</td>
</tr>
<tr>
<td>OrgsPeople</td>
<td>68.22±1.23</td>
<td>70.93±1.11</td>
<td>70.71±0.62</td>
<td>71.01±0.81</td>
<td>71.98±0.62</td>
</tr>
</tbody>
</table>

From Table 2 and 3, we can see that in terms of classification performance, using RBF kernel, all the methods can improve the classification performance on all data sets. For example, in German organisers and Glass2, the classification accuracies of LPSSVM improve more than ten percent in the RBF kernel than in the linear case. LPSSVM demonstrates the highest classification accuracy in almost all of the data sets, which validates that it is more stable than SVM, MCLPV_SVM, SDA, and LapSVM. This fortifies that the discriminative information can be obtained from the local manifold structure presented by the within-class locality preserving scatter. Meanwhile LPSSVM fully utilizes the distribution of unlabeled samples, so its decision hyperplane will not traverse the unlabeled samples like SVM, MCLPV_SVM do sometimes.

As can be seen in Figure 3, with the increase of the number of labeled samples in the training set, the average classification accuracy of each benchmarking methods goes up steadily, and the proposed LPSSVM always outperforms other benchmarking methods. Meanwhile, with the increase of the number of labeled samples, the average classification accuracy of LPSSVM has been enhanced greater than LapSVM. These experimental results indeed manifest our conclusion in the previous sections that since the LPSSVM consider both underlying discriminative information and the local geometry of the samples, it can achieve better performance.

As can be seen in Figure 4, as the supervised methods, the training time of SVM and MCLPV_SVM always remains unchanged respectively with the increase of unlabeled samples in the training data sets. However, the training time of other semi-supervised methods becomes longer. It can also be seen that the training time of LPSSVM and LapSVM is almost the same on all the data sets.
Figure 3. Comparisons of average classification accuracy with standard deviation on different data sets with different labeled samples.
4.4 Experiments of different face recognition data sets

The face recognition data sets usually have an intrinsic invariant and associated transformation. That is, these data naturally imply a manifold on which those neighboring points are small transformation of one another [32]. Moreover, the face recognition data sets have obvious non-linear manifolds structure. Therefore, in this section, we apply our method in the nonlinear case in the context of face recognition. In order to further investigate the effectiveness of the proposed LPSSVM method, we use two measures of distance: the geodesic distance and the Euclidean distance whose method is named eLPSSVM. This comparison has been carried out on the Yale face data set and the CMU PIE face data set.

The Yale face data set contains 165 images of 15 subjects, and each subject providing 11 different images under various facial expressions and lighting conditions. In our experiments, each image is resized to (32 × 32), resulting in an input dimensionality of $N=1024$. Figure 5 shows sample images of Yale. For each subject, 6 images (90 images in total)
are randomly selected as the training set. Among the 6 images per subject, 3 images are randomly selected and labeled, while leaving the other 3 images per subject unlabeled. The rest of the data set was considered to be the testing set. The CMU PIE face data set contained 68 subjects with 41,368 face images. The face images were captured by 13 synchronized cameras and 21 flashes, under varying poses, illumination and expressions. Figure 6 shows sample images of PIE. We used 170 face images for each subject in our experiment, 25 labeled and 60 unlabeled images for training and the other 85 images for testing.

First, for all these data, the kernel principal component analysis (KPCA) [33] is used to project the data in the original image space into a lower dimensional subspace. As the pattern often contains redundant information, mapping it to a feature vector can purge this redundancy while yet preserving most of the intrinsic information contents of the pattern. Then, we repeated our experiment with projecting all the images onto KPCA subspace of 10, 20, 30,…, 100 dimensions each. One-against-all algorithm is used for all the methods. We average the results over 20 random split. The experimental results are reported in Figure 7 and Figure 8.

Figure 5. Example of face images from Yale.

Figure 6. Example of face images from PIE.

Figure 7. Comparison of classification accuracy (%) vs. KPCA dimensions on Yale data set among the six methods.

Figure 8. Comparison of classification accuracy (%) vs. KPCA dimensions on PIE data set among the six methods.

Figure 9. The performance influence of LPSSVM and eLPSSVM based on different $k$ on PIE data set.

Figure 7 and Figure 8 indicate when dealing with face recognition data sets of obvious manifold structure, LPSSVM delivers more stable results across all the data sets and is highly competitive in all of the data sets. LPSSVM not only keeps local manifold structure to a certain extent but also makes sure that the within-class locality preserving scatter is minimized, thus maintaining the underlying local discriminated information of face recognition data sets. It is
noticed that in two face recognition data sets, LPSSVM outperforms eLPSSVM. Therefore it can be seen that using the geodesic distance in distance measure can improve the accuracy in image recognition.

To evaluate the influence of the neighborhood parameter $k$ on the performance, the average accuracy of experiments for data set PIE is recorded for each value of $k \in [1, 11]$. The result is showed in Figure 9. It can be seen that in LPSSVM when $k$ is larger than 4, to preserve so much local relation may be inappropriate, and the classification accuracies begin to decrease. However, with the increase of $k$, the accuracy of eLPSSVM increases, but it still below the best accuracy of LPSSVM. The reason is that the geodesic distance is approximated by finding the shortest path in a graph with edges connecting neighboring data points. When $k$ is big enough, the geodesic distance could not computed by adding up a sequence of “short hops” between neighboring points.

5 Conclusion

In this paper, we propose a novel SVM method called LPSSVM for semi-supervised classification tasks via embedding within-class locality manifold regularization. LPSSVM exploits the within-class locality manifold information of data used for semi-supervised classification by fully utilizing both discriminative information and underlying distribution of data. In general, LPSSVM takes the intrinsic manifold structure of the data space into full consideration, rather than only emphasizing the smoothness of the classifier in traditional regularization methods. Meanwhile, LPSSVM adopts geodesic distance metric to measure the distance between data in the manifold space, which can reflect the true geometry of the manifold. LPSSVM can be integrated into the classical SVM framework and implemented using the existing SVM software, which makes it more practicable for real world applications. The experimental results demonstrate that our method obtains consistently better classification accuracy than the related methods, such as the standard SVM, MCLPV_SVM, SDA and LapSVM.

In future works, we intend to perform investigations to large-scale classification problems. Moreover, adaptive parameter selection is an open problem in manifold learning and regularization.

Conflict of Interests

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

Acknowledgment

This work was supported by the National Natural Science Foundation of China under Contact (61272210, 61170122, 61202311), by the Natural Science Foundation of Jiangsu Province under Grant BK201221834, and by the R&D Frontier Grant of Jiangsu Province under Grant BY2013015-02.

References


