

Manifold proximal support vector machine for semi-supervised classification

Wei-Jie Chen · Yuan-Hai Shao · Deng-Ke Xu · Yong-Feng Fu

Published online: 21 December 2013
© Springer Science+Business Media New York 2013

Abstract Recently, semi-supervised learning (SSL) has attracted a great deal of attention in the machine learning community. Under SSL, large amounts of unlabeled data are used to assist the learning procedure to construct a more reasonable classifier. In this paper, we propose a novel manifold proximal support vector machine (MPSVM) for semi-supervised classification. By introducing discriminant information in the manifold regularization (MR), MPSVM not only introduces MR terms to capture as much geometric information as possible from inside the data, but also utilizes the maximum distance criterion to characterize the discrepancy between different classes, leading to the solution of a pair of eigenvalue problems. In addition, an efficient particle swarm optimization (PSO)-based model selection approach is suggested for MPSVM. Experimental results on several artificial as well as real-world datasets demonstrate that MPSVM obtains significantly better performance than supervised GEPSVM, and achieves comparable or better performance than LapSVM and LapTSVM, with better learning efficiency.

Keywords Semi-supervised classification · Manifold regularization · Support vector machine · Nonparallel hyperplanes · Particle swarm optimization

1 Introduction

Over the last decade, support vector machines (SVMs) have been recognized as a powerful kernel-based tool for machine learning because of their remarkable generalization performance [1–3]. In contrast with the conventional artificial neural networks, which aim to reduce the empirical risk, SVMs are guided by the principle of structural risk minimization (SRM) to guarantee the upper bound of the generalization error [1, 2]. The central idea of SVMs is to construct two optimal parallel hyperplanes that maximize the margin between two classes (data labeled as “+1” or “−1”) by solving a quadratic programming problem (QPP). Within a few years of their introduction, SVMs had already outperformed most machine learning methods in a wide variety of applications [4–8].

Recently, Mangasarian et al. [9] proposed a generalized eigenvalue proximal support vector machine (GEPSVM) for supervised classification problems. GEPSVM aims to generate two nonparallel proximal hyperplanes, with each hyperplane closer to its own class and as far as possible from the other. For this purpose, it solves a pair of relatively smaller optimization problems, instead of the large one considered by traditional SVMs [1]. As a result, the learning procedure of GEPSVM is more efficient than that of SVMs [9]. In addition, GEPSVM is excellent at dealing with “xor” problems. Thus, methods of constructing nonparallel proximal classifiers have been extensively studied, such as improved GEPSVM [10], DGEPSVM [11], TWSVM [12, 13],

W.-J. Chen (✉) · Y.-H. Shao · Y.-F. Fu
Zhijiang College, Zhejiang University of Technology,
Hangzhou 310024, P.R. China
e-mail: wjcp2008@126.com

Y.-H. Shao
e-mail: shaoyuanhai21@163.com

Y.-F. Fu
e-mail: fuyongfeng@zjc.zjut.edu.cn

D.-K. Xu
Department of Statistics, Zhejiang Agriculture and Forest
University, Lin’an 311300, P.R. China
e-mail: xudengke1983@163.com

twin parametric-margin SVM (TPMSVM) [14], structural TWSVM (S-TWSVM) [15] and so on [16–20].

The above nonparallel proximal classifiers are fully supervised, and their generalization performance is very dependent on whether there is sufficient labeled information [21, 22]. That is to say, only labeled data are considered for model training. However, in many real-world learning problems, e.g., natural language parsing [23], spam filtering [24], video surveillance [25] and protein 3D structure prediction [26], the acquisition of labeled data is usually hard or expensive, whereas the collection of unlabeled data is much easier. In such a situation, the performance of these fully supervised classifiers usually deteriorates because of an insufficient volume of labeled information.

To deal with the situation of large amounts of unlabeled data and relatively few labeled data, the paradigm of semi-supervised learning (SSL) has been proposed. Comprehensive reviews of SSL can be found in [21, 22, 27, 28]. Among these, manifold regularization (MR) is one of the most elegant constructions [29, 30]. In the MR framework, two regularization terms are introduced: one concentrates on the complexity of the classifier in the Reproducing Kernel Hilbert Spaces (RKHS), and the other enforces the smoothness of the classifier along the intrinsic manifold. Following the MR framework, Qi et al. [31] first extended the supervised nonparallel proximal classifier to the semi-supervised case and proposed a Laplacian twin support vector machine (LapTSVM). Extensive experimental results [31–33] demonstrated the effectiveness of this approach. However, one of the main challenges in LapTSVM is that the objective functions of its dual QPPs require two matrix inversion operations. These matrices are of size $(n + 1) \times (n + 1)$ for the linear case and $(l + u + 1) \times (l + u + 1)$ for the nonlinear case, where n is the feature dimension and l/u is the number of labeled/unlabeled data. To our knowledge, this matrix inversion is the main bottleneck of LapTSVM, greatly reducing its learning efficiency. Another challenge is that there are at least three predetermined parameters in LapTSVM. Although a grid-based approach can be used to optimize these parameters [31], this makes the model selection of LapTSVM something of a burden. These drawbacks restrict the application of LapTSVM to many real-world problems.

In this paper, we propose a novel nonparallel proximal classifier, termed as a manifold proximal support vector machine (MPSVM), for semi-supervised classification problems. In MPSVM, we not only introduce MR terms to capture as much geometric information as possible from inside the data, but also utilize the maximum distance criterion to characterize the discrepancy between different classes. MPSVM has the following properties:

- MPSVM determines a pair of nonparallel proximal hyperplanes by solving two standard eigenvalue problems, successfully avoiding the matrix inversion operations.

- An efficient particle swarm optimization (PSO)-based model selection (parameter optimization) approach is designed for MPSVM. By applying PSO, our MPSVM achieves better learning efficiency.
- MPSVM has a natural out-of-sample extension property from training data to unseen data. This can handle both the transductive and inductive learning cases.
- Finally, by choosing an appropriate parameter, MPSVM can degenerate to a supervised nonparallel proximal classifier, i.e., GEPSVM [9] and DGEPSVM [11].

The remainder of this paper is organized as follows: In Sect. 2, a brief review of SVM and GEPSVM is given. Our linear and nonlinear MPSVM is formulated in Sect. 3. The relations between MPSVM and some other related methods are also discussed in Sect. 3. In Sect. 4, PSO-based model selection approach for MPSVM is arranged. Experimental results are described in Sect. 5 and concluding remarks are given in Sect. 6.

2 Preliminaries

In this paper, all vectors are column vectors unless transformed to a row vector by a prime superscript $'$. A vector of zeros of arbitrary dimension is represented by $\mathbf{0}$. In addition, we denote \mathbf{e} as a vector of ones of arbitrary dimension and \mathbf{I} as an identity matrix of arbitrary dimensions.

2.1 Support vector machine

As a state-of-the-art of supervised machine learning method, support vector machine (SVM) [1, 3] has been introduced under the framework of statistical learning theory, which is known as the SRM principle. Consider a binary classification problem in the n -dimensional real space \mathbb{R}^n . Given a set of labeled data $\mathbf{T} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_l, y_l)\}$, where $\mathbf{X}_l = \{\mathbf{x}_i\}_{i=1}^l \in \mathbb{R}^{l \times n}$ are inputs and $\mathbf{Y}_l = \{y_i\}_{i=1}^l \in \{1, -1\}^l$ are corresponding labels. SVM aims to maximize the margin between two different classes by constructing the following separating hyperplane:

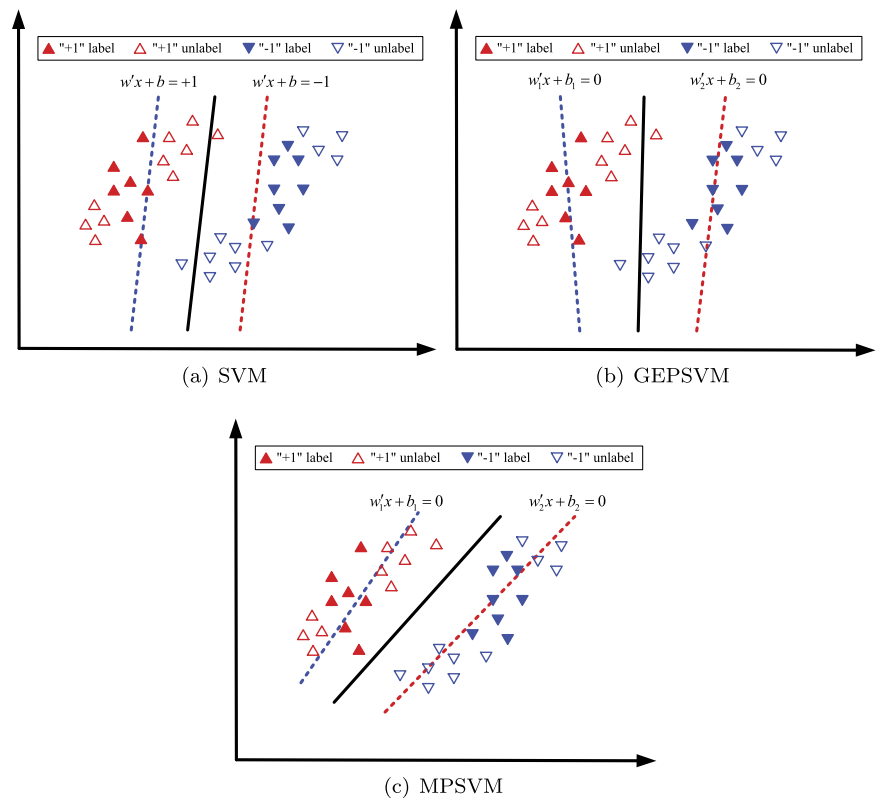
$$f(\mathbf{x}) : \mathbf{w}'\mathbf{x} + b = 0, \quad (1)$$

where $\mathbf{w} \in \mathbb{R}^n$ is the normal vector and $b \in \mathbb{R}$ is the bias term. Then, the hyperplane (1) is obtained by solving the following QPP:

$$\begin{aligned} \min_{\mathbf{w}_1, b_1, \xi} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + c\mathbf{e}'\xi, \\ \text{s.t.} \quad & \mathbf{Y}_l(\mathbf{X}_l\mathbf{w} + \mathbf{e}b) + \xi \geq \mathbf{e}, \quad \xi \geq \mathbf{0}, \end{aligned} \quad (2)$$

where $\|\cdot\|$ stands for the L_2 -norm, $\xi \in \mathbb{R}^l$ are the slack variables, $c > 0$ is the regularization factor that balances the importance between the maximization of the margin and the

Fig. 1 Geometric interpretation of SVM, GEPSVM and MPSVM on the toy example (Color figure online)



minimization of the empirical risks. An intuitive geometric interpretation for the linear SVM is shown in Fig. 1(a).

Note that the minimization of the regularization term $\frac{1}{2}\|\mathbf{w}\|^2$ is equivalent to the maximization of the margin between two parallel hyperplanes $\mathbf{w}'\mathbf{x} + b = 1$ and $\mathbf{w}'\mathbf{x} + b = -1$. When we obtain the optimal solution of (2), a new point $\mathbf{x} \in \mathbb{R}^n$ is classified as “+1” or “-1” according to whether the decision function,

$$\text{Class } i = \text{sign}(\mathbf{w}'\mathbf{x} + b), \tag{3}$$

yields “+1” or “-1”.

2.2 Generalized eigenvalue proximal SVM

Generalized eigenvalue proximal SVM (GEPSVM) is one of the most well-known supervised nonparallel proximal classifiers. Let us denote $\mathbf{A} \in \mathbb{R}^{m_1 \times n}$ as the labeled data belonging to “+1” class, and $\mathbf{B} \in \mathbb{R}^{m_2 \times n}$ as the labeled data belonging to “-1” class, where $m_1 + m_2 = l$. The original idea of GEPSVM [9] is to seek the following two nonparallel proximal hyperplanes¹

$$f_1(\mathbf{x}) : \mathbf{w}'_1\mathbf{x} + b_1 = 0 \quad \text{and} \quad f_2(\mathbf{x}) : \mathbf{w}'_2\mathbf{x} + b_2 = 0, \tag{4}$$

¹We use b_1 and b_2 instead of $-\gamma_1$ and $-\gamma_2$ in the original paper [9] only for the unified notation.

where $\mathbf{w}_1, \mathbf{w}_2 \in \mathbb{R}^n$ are the normal vectors and $b_1, b_2 \in \mathbb{R}$ are the bias terms, each hyperplane is closer to its class and is as far as possible from the other. Then, the optimization problems for GEPSVM can be expressed as

$$\min_{(\mathbf{w}_1, b_1) \neq \mathbf{0}} \frac{\|\mathbf{A}\mathbf{w}_1 + \mathbf{e}_1 b_1\|^2 / \|\begin{bmatrix} \mathbf{w}_1 \\ b_1 \end{bmatrix}\|^2}{\|\mathbf{B}\mathbf{w}_1 + \mathbf{e}_2 b_1\|^2 / \|\begin{bmatrix} \mathbf{w}_1 \\ b_1 \end{bmatrix}\|^2}, \tag{5}$$

and

$$\min_{(\mathbf{w}_2, b_2) \neq \mathbf{0}} \frac{\|\mathbf{B}\mathbf{w}_2 + \mathbf{e}_2 b_2\|^2 / \|\begin{bmatrix} \mathbf{w}_2 \\ b_2 \end{bmatrix}\|^2}{\|\mathbf{A}\mathbf{w}_2 + \mathbf{e}_1 b_2\|^2 / \|\begin{bmatrix} \mathbf{w}_2 \\ b_2 \end{bmatrix}\|^2}. \tag{6}$$

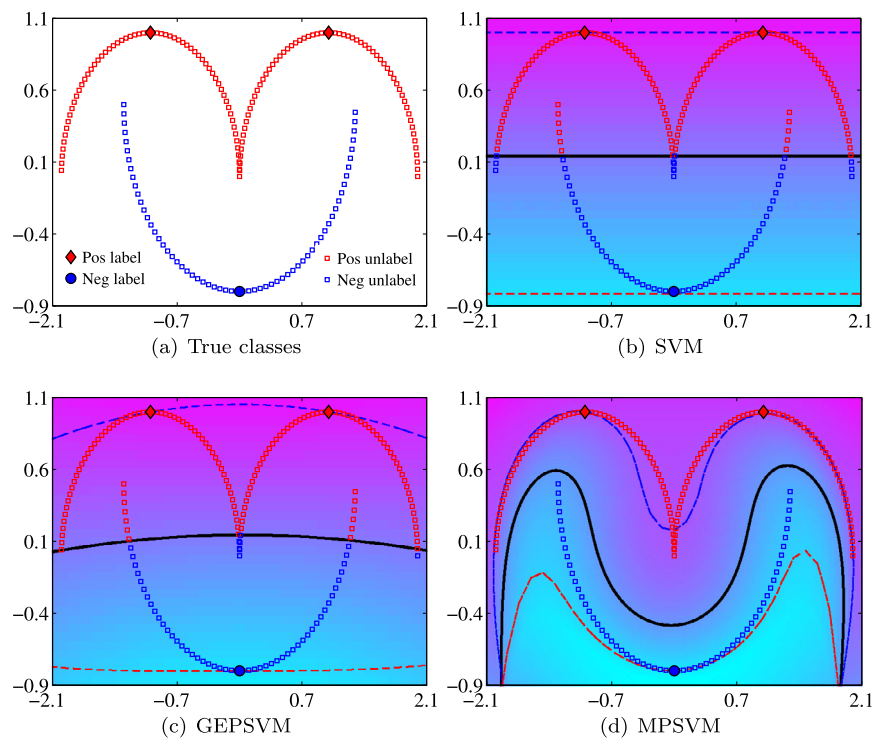
In order to reduce the norm problem of variables (\mathbf{w}_i, b_i) ($i = 1, 2$) in (5) and (6), GEPSVM introduces a Tikhonov regularization term [9, 34] and further regularizes the optimization problems as

$$\min_{(\mathbf{w}_1, b_1) \neq \mathbf{0}} \frac{\|\mathbf{A}\mathbf{w}_1 + \mathbf{e}_1 b_1\|^2 + \delta \|\begin{bmatrix} \mathbf{w}_1 \\ b_1 \end{bmatrix}\|^2}{\|\mathbf{B}\mathbf{w}_1 + \mathbf{e}_2 b_1\|^2}, \tag{7}$$

and

$$\min_{(\mathbf{w}_2, b_2) \neq \mathbf{0}} \frac{\|\mathbf{B}\mathbf{w}_2 + \mathbf{e}_2 b_2\|^2 + \delta \|\begin{bmatrix} \mathbf{w}_2 \\ b_2 \end{bmatrix}\|^2}{\|\mathbf{A}\mathbf{w}_2 + \mathbf{e}_1 b_2\|^2}, \tag{8}$$

Fig. 2 Synthetic smile datasets without noise. The *upper part* corresponds to positive class, and the *lower part* corresponds to negative class. The *squares* denote a large set of unlabeled data points. The *red diamond* or *blue circle* denotes the labeled data points of positive or negative class, respectively. The *black solid curve* is the decision boundary. The *blue* and *red dashed curves* are the two kernel-generated hyperplanes. The nonlinear classification accuracy of SVM 87.83 %, GEPSVM 89.41 %, and MPSVM 100.00 % (Color figure online)



where $\delta > 0$ is the regularization parameter. An intuitive geometric interpretation for the linear GEPSVM is shown in Fig. 1(b).

By defining $\mathbf{G} = [\mathbf{A} \mathbf{e}_1]'[\mathbf{A} \mathbf{e}_1] + \delta \mathbf{I}$, $\mathbf{H} = [\mathbf{B} \mathbf{e}_2]'[\mathbf{B} \mathbf{e}_2]$, $\mathbf{L} = [\mathbf{B} \mathbf{e}_2]'[\mathbf{B} \mathbf{e}_2] + \delta \mathbf{I}$, $\mathbf{M} = [\mathbf{A} \mathbf{e}_1]'[\mathbf{A} \mathbf{e}_1]$, $\mathbf{v}_1 = \begin{bmatrix} w_1 \\ b_1 \end{bmatrix}$ and $\mathbf{v}_2 = \begin{bmatrix} w_2 \\ b_2 \end{bmatrix}$, we can reformulate (7) and (8) as

$$\min_{\mathbf{v}_1 \neq \mathbf{0}} \frac{\mathbf{v}_1' \mathbf{G} \mathbf{v}_1}{\mathbf{v}_1' \mathbf{H} \mathbf{v}_1} \quad \text{and} \quad \min_{\mathbf{v}_2 \neq \mathbf{0}} \frac{\mathbf{v}_2' \mathbf{L} \mathbf{v}_2}{\mathbf{v}_2' \mathbf{M} \mathbf{v}_2}. \quad (9)$$

According to [9, 35], the above two minimization problems are exactly Rayleigh quotient and the solutions can be readily computed by solving the following two related generalized eigenvalue problems (GEPs)

$$\mathbf{G} \mathbf{v}_1 = \lambda_1 \mathbf{H} \mathbf{v}_1 \quad \text{and} \quad \mathbf{L} \mathbf{v}_2 = \lambda_2 \mathbf{M} \mathbf{v}_2. \quad (10)$$

Specially, the eigenvectors of (10) corresponding to the smallest eigenvalues are the optimal solutions to (7) and (8). Once the solutions (\mathbf{w}_1, b_1) and (\mathbf{w}_2, b_2) are obtained, a new point $\mathbf{x} \in \mathbb{R}^n$ is assigned to class i ($i = "+1"$ or $"-1"$), depending on which of the two hyperplanes (4) it lies closer to, i.e.,

$$\text{Class } i = \arg \min_{k=1,2} \frac{|\mathbf{w}_k' \mathbf{x} + b_k|}{\|\mathbf{w}_k\|}, \quad (11)$$

where $|\cdot|$ is the absolute value.

3 Manifold proximal SVM

3.1 Motivation

Let us denote $\mathbf{X}_u = \{\mathbf{x}_i\}_{i=l+1}^{l+u} \in \mathbb{R}^{u \times n}$ as the unlabeled data, and $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{l+u} \in \mathbb{R}^{(l+u) \times n}$ as all the training data. As mentioned previously, the optimization problems in both SVM and GEPSVM only consider the labeled data \mathbf{X}_l , but omit the distribution information revealed by the unlabeled data \mathbf{X}_u . Therefore, their performance will deteriorate when the amount of labeled information is insufficient.

For example, imagine a situation where three labeled data (two positive and one negative) and some unlabeled data are given, as illustrated in Fig. 2(a). If a classifier is constructed using only these three labeled data, an optimal choice appears to be the "mid" hyperplane between them. As a result, SVM and GEPSVM cannot capture the real data distribution/tendency, which is shown in Figs. 2(b) and (c).

Thus, to make full use of both the labeled data \mathbf{X}_l and unlabeled data \mathbf{X}_u , we primarily propose a novel manifold proximal SVM (MPSVM) for semi-supervised classification problems. Inspired by the maximum distance criterion [9–11] and MR technique [29, 30], our MPSVM incorporates both discriminant information and distribution information by minimizing the following two optimization problems

$$f_1^* = \arg \left(\begin{array}{l} \min_{f_1 \in \mathcal{H}} \quad R^{emp}(f_1(\mathbf{X}_l)) + \gamma_{\mathcal{M}} \|\mathbf{f}_1(\mathbf{X})\|_{\mathcal{M}}^2 \\ \text{s.t.} \quad \|\mathbf{f}_1\|_{\mathcal{H}}^2 = 1 \end{array} \right), \quad (12)$$

and

$$f_2^* = \arg \left(\begin{array}{l} \min_{f_2 \in \mathcal{H}} R^{emp}(f_2(\mathbf{X}_l)) + \gamma_{\mathcal{M}} \|f_2(\mathbf{X})\|_{\mathcal{M}}^2, \\ \text{s.t.} \quad \|f_2\|_{\mathcal{H}}^2 = 1 \end{array} \right), \tag{13}$$

where $R^{emp}(f)$ denotes the empirical risks on the labeled data \mathbf{X}_l , which are used to extract the discriminant information for MPSVM. In light of the manifold assumption that two points $\mathbf{x}_1, \mathbf{x}_2$ that are close on the intrinsic manifold \mathcal{M} should share similar labels, the MR term $\|f\|_{\mathcal{M}}^2$ enforces the smoothness of f along the underlying distribution (intrinsic manifold \mathcal{M}). Moreover, $\|f\|_{\mathcal{H}}^2$ is the norm of f in the RKHS, and the constraint controls the complexity of MPSVM to avoid over-fitting. In the following subsections, we will give the derivation of these terms in (12) and (13) for both linear and nonlinear cases.

3.2 Linear MPSVM

For the linear case, our MPSVM finds the following two nonparallel proximal hyperplanes

$$f_1(\mathbf{x}) : \mathbf{w}'_1 \mathbf{x} + b_1 = 0 \quad \text{and} \quad f_2(\mathbf{x}) : \mathbf{w}'_2 \mathbf{x} + b_2 = 0, \tag{14}$$

where $\mathbf{w}_1, \mathbf{w}_2 \in \mathbb{R}^n$ are the normal vectors and $b_1, b_2 \in \mathbb{R}$ are the bias terms.

Motivated by the maximum distance criterion,² we use the “difference” instead of the “ratio” (used in GEPSVM) to characterize the discrepancy between two different classes. Thus, the empirical risk $R^{emp}(f)$ in (12) and (13) can be represented as

$$R^{emp}(f_1) = \|\mathbf{A}\mathbf{w}_1 + \mathbf{e}_1 b_1\|^2 / \left\| \begin{bmatrix} \mathbf{w}_1 \\ b_1 \end{bmatrix} \right\|^2 - c_1 \|\mathbf{B}\mathbf{w}_1 + \mathbf{e}_2 b_1\|^2 / \left\| \begin{bmatrix} \mathbf{w}_1 \\ b_1 \end{bmatrix} \right\|^2, \tag{15}$$

and

$$R^{emp}(f_2) = \|\mathbf{B}\mathbf{w}_2 + \mathbf{e}_2 b_2\|^2 / \left\| \begin{bmatrix} \mathbf{w}_2 \\ b_2 \end{bmatrix} \right\|^2 - c_1 \|\mathbf{A}\mathbf{w}_2 + \mathbf{e}_1 b_2\|^2 / \left\| \begin{bmatrix} \mathbf{w}_2 \\ b_2 \end{bmatrix} \right\|^2, \tag{16}$$

where $c_1 > 0$ is the empirical risk penalty parameter that determines the trade-off between the two terms in (15) and (16). That is to say, introducing the parameter c_1 allows our MPSVM to have a bias factor for different data classes.

²According to [10, 11], using the “difference” instead of the “ratio” does not change the geometrical interpretation of GEPSVM, results in standard eigenvalue problems, which are more efficient than the general eigenvalue problems solved in GEPSVM. Moreover, comprehensive comparisons in [10, 11] show that the “difference” has comparable or better performance compared to the “ratio” (GEPSVM), but with the less learning time.

Generally, in SSL [29, 30], the MR terms $\|f\|_{\mathcal{M}}^2$ can be approximated by

$$\begin{aligned} \|f_1\|_{\mathcal{M}}^2 &= \sum_{i,j=1}^{l+u} w_{ij} (f_1(\mathbf{x}_i) - f_1(\mathbf{x}_j))^2 = f_1'(\mathbf{X}) \mathbf{L} f_1(\mathbf{X}) \\ &= (\mathbf{X}\mathbf{w}_1 + \mathbf{e}b_1)' \mathbf{L} (\mathbf{X}\mathbf{w}_1 + \mathbf{e}b_1), \end{aligned} \tag{17}$$

and

$$\begin{aligned} \|f_2\|_{\mathcal{M}}^2 &= \sum_{i,j=1}^{l+u} w_{ij} (f_2(\mathbf{x}_i) - f_2(\mathbf{x}_j))^2 = f_2'(\mathbf{X}) \mathbf{L} f_2(\mathbf{X}) \\ &= (\mathbf{X}\mathbf{w}_2 + \mathbf{e}b_2)' \mathbf{L} (\mathbf{X}\mathbf{w}_2 + \mathbf{e}b_2), \end{aligned} \tag{18}$$

where w_{ij} is the edge-weight defined for a pair of points $(\mathbf{x}_i, \mathbf{x}_j)$ of the adjacency matrix $\mathbf{W} = (w_{ij}) \in \mathbb{R}^{(l+u) \times (l+u)}$, $f_1(\mathbf{X}) = \mathbf{X}\mathbf{w}_1 + \mathbf{e}b_1$, $f_2(\mathbf{X}) = \mathbf{X}\mathbf{w}_2 + \mathbf{e}b_2$, and \mathbf{L} is the graph Laplacian defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$. Furthermore, the diagonal matrix \mathbf{D} is given by $D_{ii} = \sum_{j=1}^{l+u} w_{ij}$. More details can be seen in [29].

Similar to [9, 10, 31], we introduce a constraint to control and normalize the norm of the problem variables (w_i, b_i) ($i = 1, 2$). By defining $\mathbf{H} = [\mathbf{A} \ \mathbf{e}_1]$, $\mathbf{G} = [\mathbf{B} \ \mathbf{e}_2]$, $\mathbf{J} = [\mathbf{X} \ \mathbf{e}]$, $\mathbf{v}_1 = \begin{bmatrix} \mathbf{w}_1 \\ b_1 \end{bmatrix}$, and $\mathbf{v}_2 = \begin{bmatrix} \mathbf{w}_2 \\ b_2 \end{bmatrix}$, the primal problems for our MPSVM can be expressed as

$$\begin{aligned} \min_{\mathbf{v}_1} \quad & f_{(1,obj)}(\mathbf{v}_1) = \mathbf{v}'_1 \mathbf{H}' \mathbf{H} \mathbf{v}_1 - c_1 \mathbf{v}'_1 \mathbf{G}' \mathbf{G} \mathbf{v}_1 \\ & \quad \quad \quad + c_2 \mathbf{v}'_1 \mathbf{J}' \mathbf{L} \mathbf{J} \mathbf{v}_1, \\ \text{s.t.} \quad & \|\mathbf{v}_1\|^2 = 1, \end{aligned} \tag{19}$$

and

$$\begin{aligned} \min_{\mathbf{v}_2} \quad & f_{(2,obj)}(\mathbf{v}_2) = \mathbf{v}'_2 \mathbf{G}' \mathbf{G} \mathbf{v}_2 - c_1 \mathbf{v}'_2 \mathbf{H}' \mathbf{H} \mathbf{v}_2 \\ & \quad \quad \quad + c_2 \mathbf{v}'_2 \mathbf{J}' \mathbf{L} \mathbf{J} \mathbf{v}_2, \\ \text{s.t.} \quad & \|\mathbf{v}_2\|^2 = 1, \end{aligned} \tag{20}$$

where $c_2 > 0$ is the MR parameter. An intuitive geometric interpretation for the linear MPSVM is shown in Fig. 1(c). Let us give a detailed explanation of the optimization problem in (19). The first term in the objective function of (19) minimizes the squared sum of values of \mathbf{A} (the data labeled “+1”) on $f_1(\mathbf{x})$, which makes the labeled data \mathbf{A} be as close as possible to the “+1” proximal hyperplane $f_1(\mathbf{x})$. Optimizing the second term leads to \mathbf{B} (the data labeled “−1”) being as far as possible from $f_1(\mathbf{x})$. It is noteworthy that the first and second terms in (19) integrate the supervised information into MPSVM according to the maximum distance criterion. The third term exploits the underlying distribution between the labeled and unlabeled data. Minimizing this term enforces the smoothness of $f_1(\mathbf{x})$ along the intrinsic manifold. The constraint in (19) controls the model complexity of $f_1(\mathbf{x})$ to avoid over-fitting.

Because the optimization problem in (20) is similar to that in (19), we mainly focus on the solution of (19). Constructing the Lagrange function of (19) with the multiplier λ_1 , gives

$$L(v_1, \lambda_1) = v_1' H' H v_1 - c_1 v_1' G' G v_1 + c_2 v_1' J' L J v_1 - \lambda_1 (\|v_1\|^2 - 1). \tag{21}$$

Setting the partial derivatives of v_1 in (21) equal to zero, we obtain

$$\nabla_{v_1} L = 2(H' H - c_1 G' G + c_2 J' L J) v_1 - 2\lambda_1 v_1 = 0, \tag{22}$$

which is equal to

$$(H' H - c_1 G' G + c_2 J' L J) v_1 = \lambda_1 v_1. \tag{23}$$

In fact, λ_1 is an eigenvalue of the symmetric matrix $(H' H - c_1 G' G + c_2 J' L J)$. In particular, we can rewrite the objective function $f_{(1,obj)}(v)$ in (19) as

$$f_{(1,obj)}(v) = v_1' H' H v_1 - c_1 v_1' G' G v_1 + c_2 v_1' J' L J v_1 = v_1' (H' H - c_1 G' G + c_2 J' L J) v_1. \tag{24}$$

Then, substituting (23) into (24), we obtain

$$\begin{aligned} f_{(1,obj)}(v_1) &= v_1' \lambda_1 v_1 = \lambda_1 \|v_1\|^2 = \lambda_1 \geq \lambda_{(1,s)} \\ &= v_{(1,s)}' \lambda_{(1,s)} v_{(1,s)} = f_{(1,obj)}(v_{(1,s)}) \\ &= f_{(1,obj)}(v_1^*), \end{aligned} \tag{25}$$

where $\lambda_{(1,s)}$ is the smallest eigenvalue of (23) and $v_{(1,s)}$ is the corresponding eigenvector. From (25), we can conclude that the eigenvector corresponding to the smallest eigenvalue of (23) is the optimal solution of (19).

In a similar way, we can find the solution of the optimization problem (20) by solving the following standard eigenvalue problem:

$$(G' G - c_1 H' H + c_2 J' L J) v_2 = \lambda_2 v_2, \tag{26}$$

where the optimal solution is the eigenvector corresponding to the smallest eigenvalue.

Once solutions (w_1, b_1) and (w_2, b_2) have been obtained by solving the two eigenvalue problems of (23) and (26), a new data $x \in \mathbb{R}^n$ is assigned to class i ($i = +1$ or -1), depending on which of the two proximal hyperplanes (14) it lies closer to, i.e.,

$$\text{Class } i = \arg \min_{k=1,2} \frac{|w_k' x + b_k|}{\|w_k\|}. \tag{27}$$

3.3 Nonlinear MPSVM

In order to extend our model to the nonlinear case, we consider the following two kernel-generated proximal hyperplanes

$$\begin{aligned} f_1(x) &: \mathcal{K}(x', X') w_1 + b_1 = 0 \quad \text{and} \\ f_2(x) &: \mathcal{K}(x', X') w_2 + b_2 = 0, \end{aligned} \tag{28}$$

where $X \in \mathbb{R}^{(l+u) \times n}$ denotes all the training data and $\mathcal{K}(\cdot, \cdot)$ is an appropriately chosen kernel, such as the radial basis function (RBF) kernel $\mathcal{K}(u, v) = e^{-\gamma \|u-v\|^2}$, $\gamma > 0$. The optimization problems for the nonlinear MPSVM can be expressed as

$$\begin{aligned} \min_{w_1, b_1} & \quad \|\mathcal{K}(A, X') w_1 + e_1 b_1\|^2 \\ & \quad - c_1 \|\mathcal{K}(B, X') w_1 + e_2 b_1\|^2 \\ & \quad + c_2 (\mathbf{K} w_1 + e b_1)' L (\mathbf{K} w_1 + e b_1), \\ \text{s.t.} & \quad \|w_1\|^2 + b_1^2 = 1, \end{aligned} \tag{29}$$

and

$$\begin{aligned} \min_{w_2, b_2} & \quad \|\mathcal{K}(B, X') w_2 + e_2 b_2\|^2 \\ & \quad - c_1 \|\mathcal{K}(A, X') w_2 + e_1 b_2\|^2 \\ & \quad + c_2 (\mathbf{K} w_2 + e b_2)' L (\mathbf{K} w_2 + e b_2), \\ \text{s.t.} & \quad \|w_2\|^2 + b_2^2 = 1, \end{aligned} \tag{30}$$

where \mathbf{K} denotes $\mathcal{K}(X, X')$, $c_1 > 0$ is the empirical risk penalty parameter, $c_2 > 0$ is the manifold regularization parameter, and L is the graph Laplacian.

By defining $H_\varphi = [\mathcal{K}(A, X') \ e_1]$, $G_\varphi = [\mathcal{K}(B, X') \ e_2]$, $J_\varphi = [\mathbf{K} \ e]$, $v_1 = \begin{bmatrix} w_1 \\ b_1 \end{bmatrix}$ and $v_2 = \begin{bmatrix} w_2 \\ b_2 \end{bmatrix}$, the above problems can be rewritten as

$$\begin{aligned} \min_{v_1} & \quad v_1' H_\varphi' H_\varphi v_1 - c_1 v_1' G_\varphi' G_\varphi v_1 + c_2 v_1' J_\varphi' L J_\varphi v_1, \\ \text{s.t.} & \quad \|v_1\|^2 = 1, \end{aligned} \tag{31}$$

and

$$\begin{aligned} \min_{v_2} & \quad v_2' G_\varphi' G_\varphi v_2 - c_1 v_2' H_\varphi' H_\varphi v_2 + c_2 v_2' J_\varphi' L J_\varphi v_2, \\ \text{s.t.} & \quad \|v_2\|^2 = 1. \end{aligned} \tag{32}$$

Similar to the linear case, the solutions of the optimization problem (31) and (32) can be obtained by solving the following two standard eigenvalue problems:

$$(H_\varphi' H_\varphi - c_1 G_\varphi' G_\varphi + c_2 J_\varphi' L J_\varphi) v_1 = \lambda_1 v_1, \tag{33}$$

and

$$(G_\varphi' G_\varphi - c_1 H_\varphi' H_\varphi + c_2 J_\varphi' L J_\varphi) v_2 = \lambda_2 v_2. \tag{34}$$

where the optimal solutions are the eigenvectors corresponding to the smallest eigenvalues.

Table 1 Summary of parameters in MPSVM

| Parameter | Description | Range | Type |
|-----------|---|----------------|--------|
| c_1 | Penalty parameter for the empirical risk | $[2^{-7}-2^5]$ | Double |
| c_2 | Penalty parameter for the manifold regularization | $[2^{-7}-2^5]$ | Double |
| γ | RBF kernel parameter (for the nonlinear case) | $[2^{-3}-2^6]$ | Double |

Once the solutions (\mathbf{w}_1, b_1) and (\mathbf{w}_2, b_2) of (31) and (32) are obtained, a new data $\mathbf{x} \in \mathbb{R}^n$ is assigned to class i ($i = "+1"$ or $"-1"$), depending on which of the two kernel-generated proximal hyperplanes (28) it lies closer to, i.e.,

$$\text{Class } i = \arg \min_{k=1,2} \frac{|\mathcal{K}(\mathbf{x}', \mathbf{X}')\mathbf{w}_k + b_k|}{\sqrt{\mathbf{w}'_k \mathbf{K} \mathbf{w}_k}}. \tag{35}$$

3.4 Relationship with some other related methods

3.4.1 Relationship with GEPSVM and DGEPSVM

As mentioned above, in (5) and (6), GEPSVM uses a ‘‘ratio’’ to quantify the discrepancy between two different classes, resulting in the generalized eigenvalue problems. To enhance its performance, DGEPSVM [11] uses the ‘‘difference’’ instead of the ‘‘ratio’’, leading to simpler optimization problems (standard eigenvalue problems). If we drop the MR terms $\|f(\mathbf{X})\|_{\mathcal{M}}^2$ in (12) and (13) by setting $c_2 = 0$, our MPSVM will degenerate to the DGEPSVM. Therefore, we can see that DGEPSVM is a special case of MPSVM. From another perspective, our MPSVM is a useful extension of GEPSVM and DGEPSVM to the semi-supervised case.

3.4.2 Relationship with LapTSVM

Both LapTSVM and MPSVM utilize information about the underlying distribution (via MR) to construct a more reasonable classifier. However, there are several obvious differences between them. First, the empirical risk $R^{emp}(f)$ in LapTSVM [31] is implemented by minimizing both the L_1 and L_2 -norm loss functions for each class, whereas, in (15) and (16), our MPSVM implements $R^{emp}(f)$ by maximizing the L_2 -norm distances between two different classes. Second, during the learning procedure, the solutions of LapTSVM are obtained by solving two QPPs with computational-costly matrix inversion operations. In contrast, our MPSVM solves two standard eigenvalue problems without matrix inversion, resulting in more effective learning ability.

4 Model selection for MPSVM

In this section, we consider the model selection (parameter optimization) for MPSVM. The parameters that should

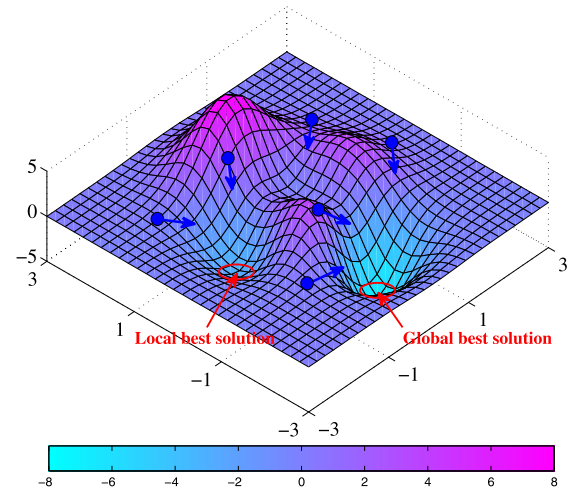


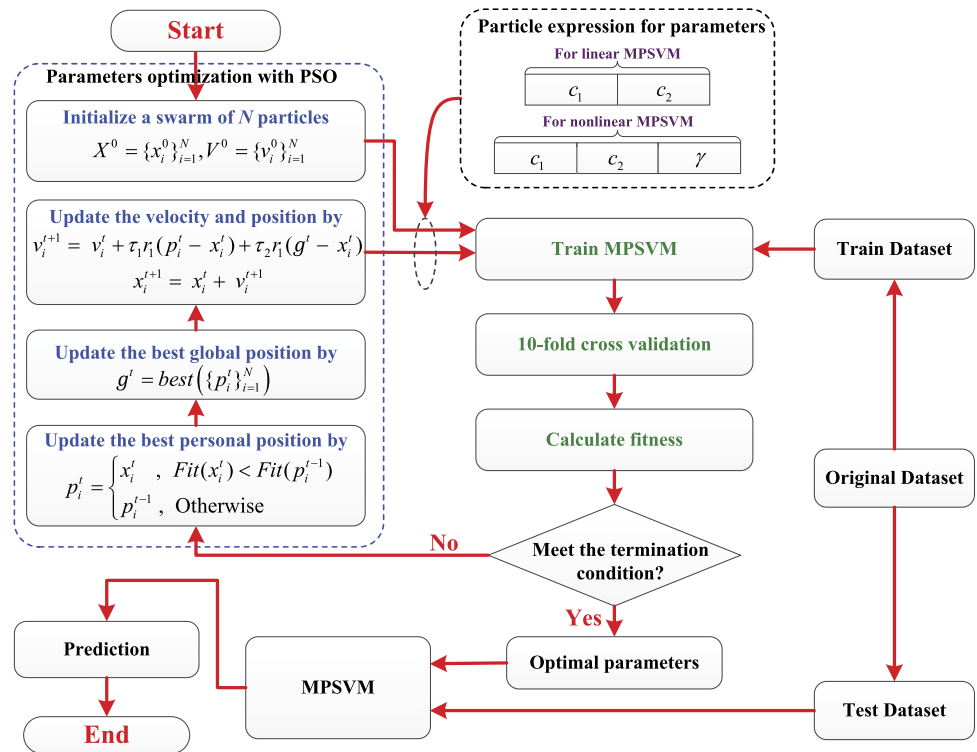
Fig. 3 The population-based search behavior of PSO. The blue circle denotes a particle and the arrow navigates the particle’s motion (search) direction (Color figure online)

be optimized in MPSVM include the empirical risk penalty parameter c_1 , the MR parameter c_2 , and the RBF kernel parameter γ (for the nonlinear case), as detailed in Table 1. Different parameter settings can have a great impact on the performance of MPSVM. However, to our knowledge, the parameter optimization is recognized as a combinatorial optimization problem (NP-hard problem), which is one of the main unsolved problems of computer science [36–38]. Typically, metaheuristics are used to obtain approximate solutions to NP-hard problems [39]. In our implementation, instead of using a genetic algorithm (GA), we apply the excellent population-based PSO metaheuristic [40, 41] to assist the parameter optimization.

4.1 Concept of particle swarm optimization (PSO)

PSO is an artificial intelligence technique that can be used to seek approximate solutions to extremely difficult numeric optimization problems [40]. Its main idea is that, inspired from the social behavior of organisms, PSO consists of a swarm (population) of particles (potential solutions) that search for the best position (solution) in the multi-dimensional space, and each particle adjusts its moving direction (velocity) according to its personal best position (cognition parts) and the best global position of all particles (social parts) during each iteration. An intuitive illustration for the population-based search behavior of PSO is shown

Fig. 4 The architecture of the proposed PSO-based parameter optimization approach for MPSVM



in Fig. 3. The iteration strategy for each particle is described as

Velocity update:

$$\mathbf{v}_i^{t+1} = \mathbf{v}_i^t + \tau_1 r_1 (\mathbf{p}_i^t - \mathbf{x}_i^t) + \tau_2 r_2 (\mathbf{g}^t - \mathbf{x}_i^t), \quad (36)$$

Position update: $\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \mathbf{v}_i^{t+1}$,

where superscript t denotes the t -th iteration; $\mathbf{v}_i = (v_{i1}, v_{i2}, \dots, v_{id})'$ and $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{id})'$ denote the velocity and position of the particle i in d -dimensional space, respectively; $\mathbf{p}_i = (p_{i1}, p_{i2}, \dots, p_{id})'$ represents the personal best position of particle i , and $\mathbf{g} = (g_1, g_2, \dots, g_d)'$ is the best position obtained from \mathbf{p}_i for all particles; $\tau_1, \tau_2 \in (0, 2]$ indicate the cognition and social learning parameter, respectively; r_1 and r_2 are the random numbers generated by uniform distribution $U[0, 1]$. More details can be seen in [40, 41].

4.2 Parameter optimization for MPSVM

In this subsection, we develop a PSO-based parameter optimization approach for our MPSVM. As indicated above, some parameters must be predetermined. These are the penalty parameters c_1, c_2 and an extra RBF kernel parameter γ for the nonlinear case. In our implementation, we first transform the above set of parameters into a particle \mathbf{x} , which is composed of (c_1, c_2) for the linear case or (c_1, c_2, γ) for the nonlinear case. The main process is illus-

trated in Fig. 4, for which we give the following explanation:

- (1) **Initialization:** A swarm of N particles is initialized to have position $\mathbf{X}^0 = \{\mathbf{x}_i^0\}_{i=1}^N$ and velocity $\mathbf{V}^0 = \{\mathbf{v}_i^0\}_{i=1}^N$. Each \mathbf{x}_i^0 and \mathbf{v}_i^0 are generated by the uniform distribution according to the range shown in Table 1. By default, the cognition learning parameter is set to $\tau_1 = 1.3$ and the social learning parameter is set to $\tau_2 = 1.5$.
- (2) **Fitness evaluation:** The fitness of each particle used to train MPSVM is evaluated according to $\text{Fit}(\mathbf{x}_i^t) = 1 - \text{Acc}(\mathbf{x}_i^t)$, where $\text{Acc}(\mathbf{x}_i^t)$ denotes the classification accuracy of MPSVM under the parameter \mathbf{x}_i^t , and the fitness $\text{Fit}(\mathbf{x}_i^t)$ denotes the corresponding training error.³
- (3) **Update operation:** If the fitness of \mathbf{x}_i^t is better than its previous best value (i.e., $\text{Fit}(\mathbf{x}_i^t) < \text{Fit}(\mathbf{p}_i^{t-1})$), the current position \mathbf{x}_i^t is taken as the new personal best position \mathbf{p}_i^t . The best $\{\mathbf{p}_i^t\}_{i=1}^N$ is then chosen as the new best global position \mathbf{g}^t . After finding the two best positions, the particle updates its velocity and position according to (36).
- (4) **Stopping criterion:** The process is terminated if the minimum error criterion is satisfied or the maximum iteration number is reached.

³A particle \mathbf{x}_i^t with higher classification accuracy produces a better fitness value (lower training error). That is, better fitness is represented by lower value.

5 Experimental results

To evaluate the performance of our MPSVM,⁴ we investigated its classification accuracy⁵ and computational efficiency⁶ on both artificial and real-world datasets. In our implementation, we focused on the comparison between MPSVM and several state-of-the-art classifiers, including GEPSVM, LapSVM, and LapTSVM:

- **GEPSVM** [9]: It is a supervised algorithm for classification. GEPSVM relaxes the universal requirement that the hyperplanes generated by SVMs should be parallel, and attempts to seek a pair of optimal nonparallel proximal hyperplanes by solving generalized eigenvalue problems. The parameter settings in GEPSVM are (δ) for linear and (δ, γ) for nonlinear.
- **LapSVM** [29]: It is an extension of SVM [1] for semi-supervised classification. LapSVM adopts the manifold assumption, and uses the hinge loss to construct a parallel hyperplane classifier by seeking a maximum margin boundary on both labeled and unlabeled data. The parameter settings in LapSVM are (c_1, c_2) for linear and (c_1, c_2, γ) for nonlinear.
- **LapTSVM** [31]: It is an extension of TWSVM [12] for semi-supervised classification. LapTSVM also adopts the manifold assumption and exploits the geometric information embedded in the training data to construct a non-parallel hyperplane classifier. The parameter settings in LapTSVM are (c_1, c_2, c_3) for linear and (c_1, c_2, c_3, γ) for nonlinear.

All the classifiers are implemented in Matlab (R14)⁷ on a personal computer (PC) with an Intel P4 processor (2.9 GHz) and 2 GB random-access memory (RAM). The general eigenvalue problem in GEPSVM and standard eigenvalue problem in MPSVM were solved by the Matlab function “eig()”. For the QPPs in LapSVM and LapTSVM, we used the Matlab function “quadprog()”. With regard to parameter selection, we employed the standard 10-fold cross-validation technique [3]. Furthermore, as in [9, 29, 31], we used a grid-based approach to obtain the optimal parameters for GEPSVM, LapSVM, and LapTSVM. For the grid-based approach, the optimal penalty parameters δ , c_1 , c_2 , c_3 and RBF kernel parameter γ were

⁴Matlab code is available at <http://www.optimal-group.org/Resource/MPSVM.html>.

⁵Classification accuracy is defined as: $Acc = \frac{TP+TN}{TP+FP+TN+FN}$, where TP, TN, FP and FN are the number of true positive, true negative, false positive and false negative, respectively.

⁶We use the training time T_{train} and parameter search time T_{para} to denote the computational efficiency (learning time) for each algorithm.

⁷Matlab is available at <http://www.mathworks.com>.

selected from the set $\{2^i | i = -5, -4, \dots, 4, 5\}$. Additionally, the PSO-based approach was utilized for our MPSVM. Once selected, the optimal parameters were employed to learn the final decision function.

5.1 Results on artificial datasets

In this subsection, we compare the effectiveness of our MPSVM with GEPSVM, LapSVM and LapTSVM for three semi-supervised artificial datasets, in terms of the classification performance and decision boundary.

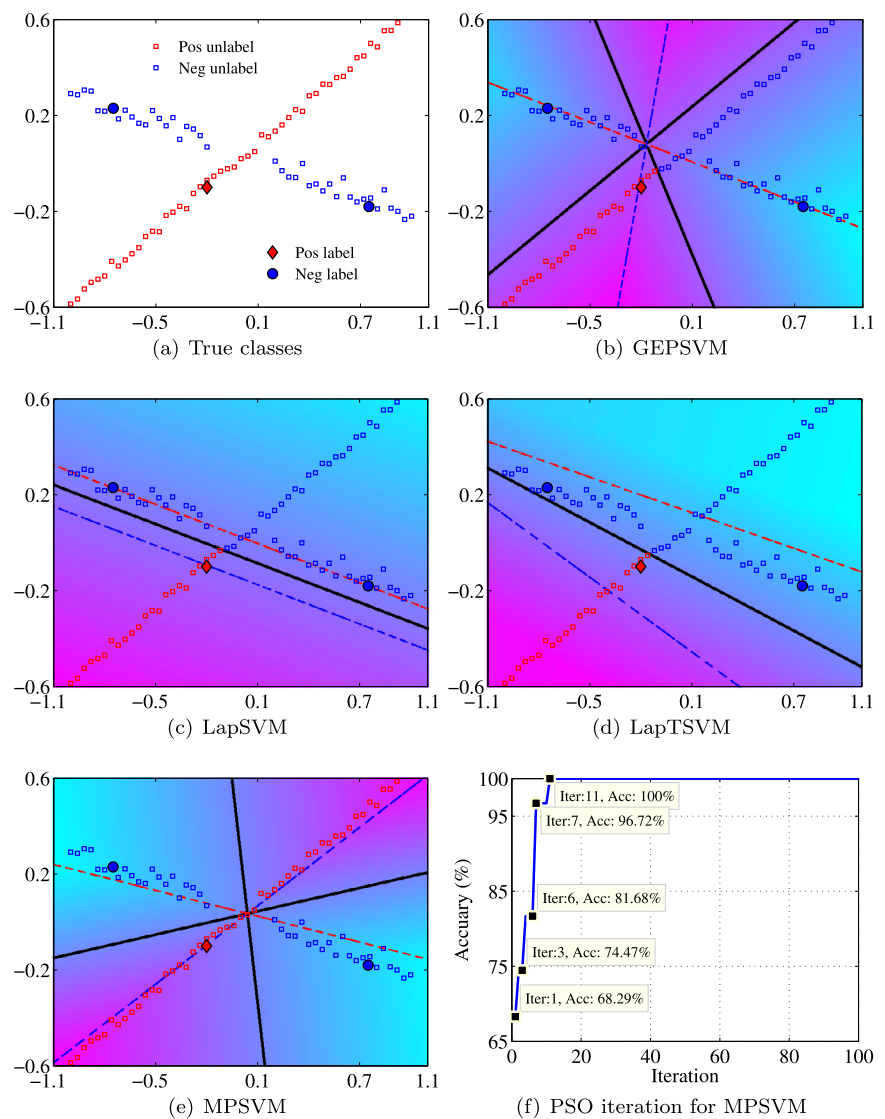
First, we consider a two-dimensional “xor” dataset, which is usually used to demonstrate the effectiveness of nonparallel proximal SVM [9, 12, 13]. The “xor” dataset was obtained by perturbing the points that lie on two intersecting planes (lines) with three labeled data points, as shown in Fig. 5(a), where each plane corresponds to one class. Figure 5 shows the one-run results from this dataset of GEPSVM, LapSVM, LapTSVM and MPSVM for the case of a linear kernel. We can see that: (1) supervised GEPSVM obtains poor results because of the insufficient labeled information; (2) although both LapSVM and LapTSVM utilize the unlabeled data to assist training, they are not suited to the “xor” dataset; (3) taking advantage of the maximum distance criterion, our MPSVM is able to deliver a more reasonable decision boundary than the others.

A more challenging case is illustrated in Fig. 6(a), which is a variant of the “smile” dataset corrupted by the Gaussian noise. Figure 6 shows the learning results of each classifiers using an RBF kernel. We can see that: (1) as might be expected, GEPSVM simply constructs the decision boundary across the minpoints of the labeled data points; (2) LapTSVM and MPSVM obtain 100 % classification accuracy. However, our MPSVM obtains a smoother decision boundary, resulting in better generalization ability.

For the third dataset (inverse half-moons, see Fig. 7(a)), we labeled three data points for each moon-shaped class. Figure 7 describes the one-run results from each classifier with this dataset using an RBF kernel. It can be seen that our MPSVM makes full use of the geometric information, and obtains a more reasonable decision boundary, whereas the other classifiers cannot achieve satisfactory performance.

We also plot the iterative PSO procedure for our MPSVM, shown in Figs. 5–7(f). We can see that the optimal model/parameters of MPSVM can be obtained after a few PSO iterations. To further illustrate the learning results of these classifiers, Table 2 lists the classification accuracies (Acc), training time (T_{train}), optimization parameters, and parameter search time (T_{para}) with these three artificial datasets. We have highlighted the best performance. The results indicate that MPSVM obtains the best classification performance among these classifiers. In terms of the training time, LapTSVM and MPSVM are more efficient than GEPSVM

Fig. 5 Synthetic xor dataset with noise. Each cross line corresponds to one class. The *squares* denote a large set of unlabeled data points. The *red diamond* or *blue circle* denotes the labeled data points of positive or negative class, respectively. The *black curve* is the decision boundary. The *blue* and *red dashed curves* are the two linear hyperplanes. The linear classification accuracy of GEPSVM 69.89 %, LapSVM 69.89 %, LapTSVM 68.82 % and MPSVM 100.00 % (Color figure online)



and LapSVM. Furthermore, the parameter search time of our PSO-based MPSVM is orders of magnitude faster than the grid-based approach.

5.2 Results on UCI datasets

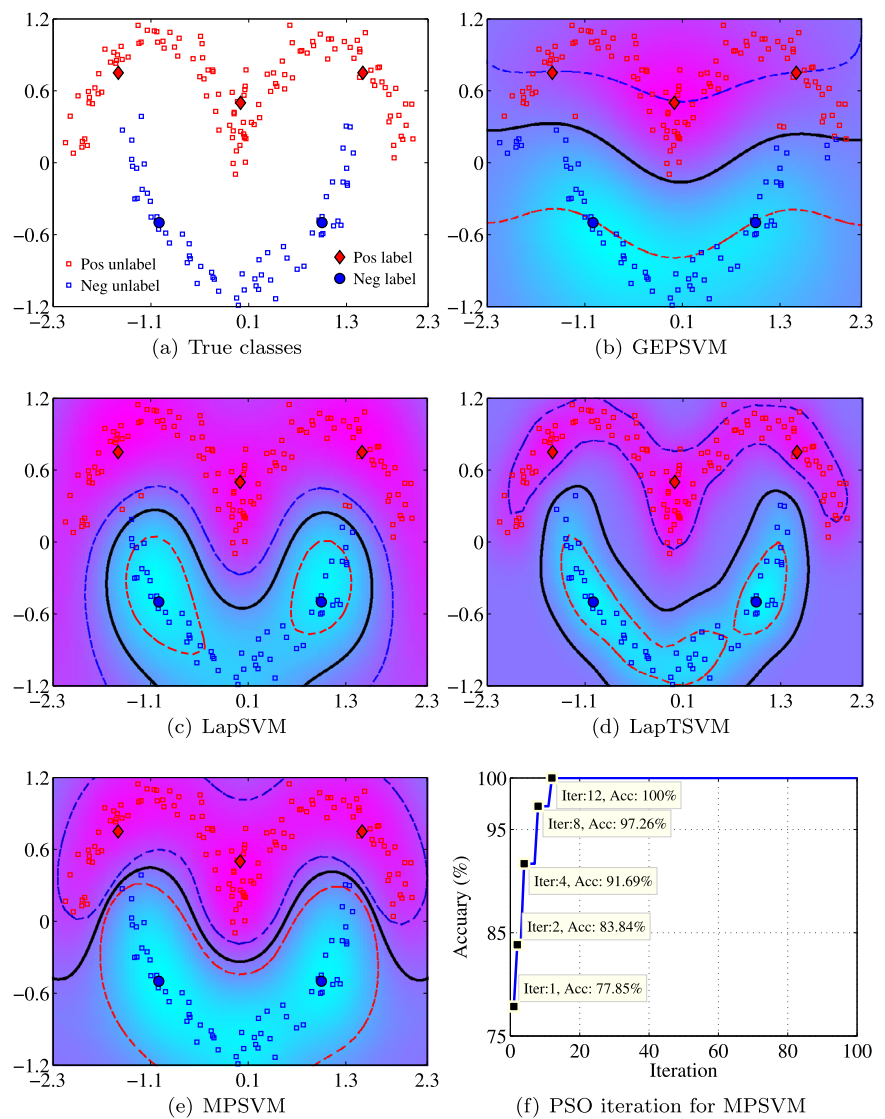
To further evaluate the performance of MPSVM, we applied each algorithm to several real-world datasets from the UCI machine learning repository,⁸ and investigated the results in terms of classification accuracy, training time, and parameter search time. We used the Hepatitis, Ionosphere, WDBC, Australian and CMC datasets for our comparison. These datasets represent a wide range of fields (include pathology, biological information, finance and so on), sizes (from 155 to 1437) and features (from 6 to 34). Note that all datasets

are normalized such that the features scale in $[-1, 1]$ before training. Similar to [27, 42], our experiments were set up in the following way. First, each dataset was divided into two subsets: 65 % for training and 35 % for testing. Then, we randomly labeled m of the training set, and used the remainder as unlabeled data, where m is the ratio of labeled data. Finally, we transformed them into semi-supervised tasks. Each experiment is repeated 10 times.

Table 3 lists the learning results of each algorithm using an RBF kernel, and includes the mean and deviation of the testing accuracy for various m from 5 % to 30 %. We have highlighted the best performance. From Table 3, it is easy to see that increasing the ratio of labeled data generally improves the classification performance for all algorithms. For example, in the Australian dataset, the accuracy of MPSVM improved more than 5 % when m increased from 5 % to 10 %. Furthermore, we also find that the traditional prox-

⁸The UCI datasets are available at <http://archive.ics.uci.edu/ml>.

Fig. 6 Synthetic smile datasets with noise. The *upper part* corresponds to positive class, and the *lower part* corresponds to negative class. The *squares* denote a large set of unlabeled data points. The *red diamond* or *blue circle* denotes the labeled data points of positive or negative class, respectively. The *black curve* is the decision boundary. The *blue* and *red dashed curves* are the two kernel-generated hyperplanes. The nonlinear classification accuracy of GEPSVM 93.65 %, LapSVM 98.17 %, LapTSVM 100.00 % and MPSVM 100.00 % (Color figure online)



imal algorithm GEPSVM performed relatively poorly with almost all datasets, which was due to insufficient numbers of labeled data. On the contrary, our MPSVM fully utilizes the underlying data information to enable better classification.

To provide more statistical evidence [27, 43], we performed a paired *t*-test to compare the testing accuracy of GEPSVM, LapSVM, and LapTSVM to that of MPSVM. The significance level (SL) was set to 0.05. That is, when the *t*-test value is greater than 1.7341, the classification results of the two algorithms significantly different. Consequently, as shown in Table 3, we can see that our MPSVM significantly outperforms GEPSVM and LapSVM with most datasets. A Win/Tie/Loss (W/T/L) summarization based on the *t*-test is also listed at the bottom of Table 3. This shows that our MPSVM obtains better classification performance than the others. This is because MPSVM combines both the

maximum distance criterion and MR to enhance its generalization ability.

The average training time (T_{train}) and parameter search time (T_{para}) of each algorithm with the above datasets are shown in Figs. 8 and 9, respectively. These reveal that the training time of MPSVM is comparable to that of LapTSVM, and the parameter search time of MPSVM is several orders of magnitude faster than that of the other classifiers.

5.3 Results for handwritten symbol recognition

In this section, we investigate the impact of the number of unlabeled data on the performance of MPSVM. The USPS handwritten dataset⁹ was used for these experiments. The

⁹The USPS datasets are available at www.cs.nyu.edu/~roweis/data.html.

Fig. 7 Synthetic two inverse half-moons datasets with noise. Each half-moon corresponds to one class. The *squares* denote a large set of unlabeled data points. The *red diamond* or *blue circle* denotes the labeled data points of positive or negative class, respectively. The *black curve* is the decision boundary. The *blue* and *red dashed curves* are the two kernel-generated hyperplanes. The nonlinear classification accuracy of GEPSVM 89.41 %, LapSVM 97.81 %, LapTSVM 98.90 % and MPSVM 99.64 % (Color figure online)

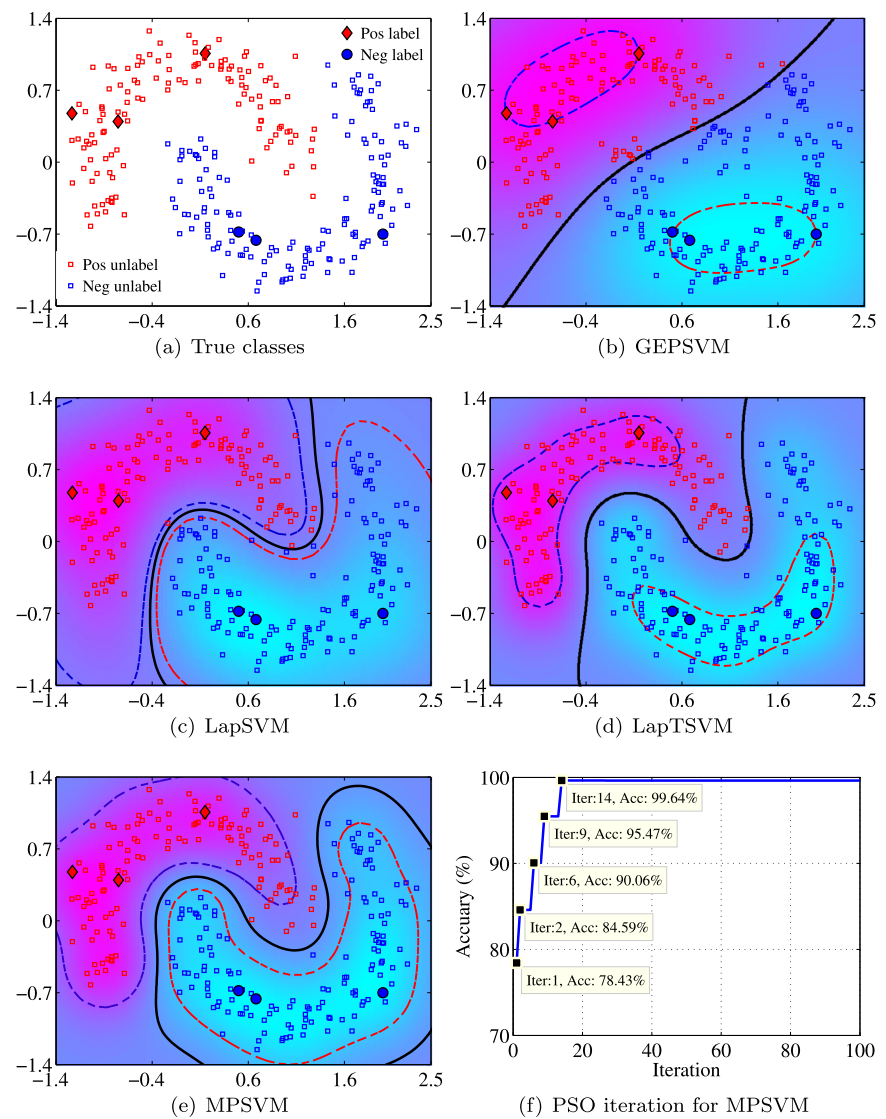


Table 2 Results of GEPSVM, LapSVM, LapTSVM and MPSVM on three artificial datasets

| Datasets | Algorithm | Acc (%) | T_{train} (s) | Parameters | T_{para} (s) |
|--|-----------|---------------|-----------------|--------------------------------|----------------|
| Cross-line ^a ($m \times n$) | GEPSVM | 69.89 | 0.0182 | (2^{-3}) | 1.3609 |
| | LapSVM | 69.89 | 0.0296 | ($2^2, 2^0$) | 8.3180 |
| | LapTSVM | 68.82 | 0.0109 | ($2^1, 2^{-3}, 2^2$) | 13.1091 |
| | MPSVM | 100.00 | 0.0071 | (0.371, 0.539) | 0.1478 |
| Simple-noise ^b ($m \times n$) | GEPSVM | 93.65 | 0.1457 | ($2^1, 2^{-2}$) | 26.8370 |
| | LapSVM | 98.17 | 0.26782 | ($2^{-2}, 2^3, 2^{-3}$) | 526.8302 |
| | LapTSVM | 100.00 | 0.0841 | ($2^1, 2^{-3}, 2^3, 2^0$) | 1138.1428 |
| | MPSVM | 100.00 | 0.0902 | (2.530, 0.792, 0.238) | 8.7524 |
| Half-moon ^b ($m \times n$) | GEPSVM | 89.41 | 0.1794 | ($2^{-2}, 2^0$) | 37.4364 |
| | LapSVM | 97.81 | 0.3169 | ($2^{-2}, 2^1, 2^2$) | 682.4264 |
| | LapTSVM | 98.90 | 0.0918 | ($2^1, 2^{-1}, 2^3, 2^{-2}$) | 1424.6019 |
| | MPSVM | 99.64 | 0.1266 | (8.383, 2.327, 0.429) | 8.7328 |

^aLinear kernel $\mathcal{K}(\mathbf{u}, \mathbf{v}) = \mathbf{u}'\mathbf{v}$

^bRBF kernel

$\mathcal{K}(\mathbf{u}, \mathbf{v}) = e^{-\gamma\|\mathbf{u}-\mathbf{v}\|^2}$

Table 3 Mean and standard deviation of testing accuracy with different labeled ratio m on UCI dataset. \blacktriangle/∇ indicates whether MPSVM is statistically superior/inferior to the compared algorithm, according to pairwise t -test 95 % significance level. Win/Tie/Loss denotes the number of datasets where MPSVM is significance Superior/Equal/Inferior to the compared algorithm. Ave. mean and std denotes the average mean and standard deviation accuracy of each algorithm over all datasets

| Datasets | Ratio (%) | GEPSVM (%) | LapSVM (%) | LapTSVM (%) | MPSVM (%) |
|-----------------------|-----------|--------------------------------|-------------------------------|-------------------------------|---------------------|
| Hepatitis (155 × 9) | $m = 5$ | 42.74 ± 9.39 \blacktriangle | 63.82 ± 7.62 | 64.23 ± 7.12 | 64.39 ± 6.34 |
| | $m = 10$ | 55.21 ± 8.67 \blacktriangle | 68.08 ± 6.51 \blacktriangle | 69.67 ± 6.64 | 69.21 ± 6.30 |
| | $m = 20$ | 60.82 ± 7.21 \blacktriangle | 72.25 ± 6.97 \blacktriangle | 73.76 ± 6.76 | 74.63 ± 5.88 |
| | $m = 30$ | 66.72 ± 8.64 \blacktriangle | 74.62 ± 5.92 | 72.82 ± 5.68 \blacktriangle | 74.82 ± 6.52 |
| Ionosphere (351 × 34) | $m = 5$ | 47.38 ± 9.85 \blacktriangle | 63.68 ± 8.83 | 60.59 ± 7.41 \blacktriangle | 63.29 ± 7.24 |
| | $m = 10$ | 54.39 ± 8.61 \blacktriangle | 65.57 ± 6.59 \blacktriangle | 66.79 ± 7.85 | 66.93 ± 7.08 |
| | $m = 20$ | 62.20 ± 7.34 \blacktriangle | 71.58 ± 7.19 | 71.21 ± 6.62 \blacktriangle | 72.87 ± 6.57 |
| | $m = 30$ | 66.96 ± 8.59 \blacktriangle | 72.45 ± 6.88 | 73.78 ± 7.34 | 73.63 ± 6.39 |
| WDBC (569 × 30) | $m = 5$ | 43.98 ± 9.76 \blacktriangle | 64.13 ± 8.01 | 62.30 ± 7.32 \blacktriangle | 64.42 ± 8.53 |
| | $m = 10$ | 52.84 ± 9.72 \blacktriangle | 65.47 ± 7.83 \blacktriangle | 67.43 ± 7.61 | 67.72 ± 7.53 |
| | $m = 20$ | 58.39 ± 8.87 \blacktriangle | 70.38 ± 8.06 \blacktriangle | 71.73 ± 7.82 \blacktriangle | 73.05 ± 7.28 |
| | $m = 30$ | 64.92 ± 8.94 \blacktriangle | 71.40 ± 7.63 \blacktriangle | 73.09 ± 6.83 | 72.93 ± 6.62 |
| Australian (690 × 14) | $m = 5$ | 39.18 ± 12.35 \blacktriangle | 51.18 ± 9.36 \blacktriangle | 54.72 ± 8.54 ∇ | 52.87 ± 7.81 |
| | $m = 10$ | 45.85 ± 9.46 \blacktriangle | 54.62 ± 8.73 \blacktriangle | 58.51 ± 8.19 | 59.53 ± 7.49 |
| | $m = 20$ | 51.39 ± 10.28 \blacktriangle | 58.87 ± 8.08 | 57.73 ± 7.82 | 58.62 ± 7.60 |
| | $m = 30$ | 53.96 ± 9.26 \blacktriangle | 59.16 ± 7.39 \blacktriangle | 60.10 ± 8.05 | 61.58 ± 8.31 |
| CMC (1473 × 9) | $m = 5$ | 42.86 ± 13.39 \blacktriangle | 51.24 ± 8.64 | 52.89 ± 9.19 | 51.72 ± 7.38 |
| | $m = 10$ | 47.56 ± 10.61 \blacktriangle | 53.46 ± 7.84 \blacktriangle | 55.60 ± 8.11 \blacktriangle | 56.94 ± 7.20 |
| | $m = 20$ | 51.72 ± 9.73 \blacktriangle | 56.74 ± 6.88 \blacktriangle | 58.66 ± 7.59 | 58.82 ± 7.57 |
| | $m = 30$ | 52.58 ± 9.26 \blacktriangle | 56.22 ± 7.29 \blacktriangle | 60.72 ± 8.71 | 60.59 ± 6.83 |
| W/T/L | | 20/0/0 | 12/8/0 | 6/13/1 | / |
| Ave. mean | | 53.08 | 63.24 | 64.31 | 64.92 |
| Ave. std | | 9.49 | 7.61 | 7.56 | 7.12 |

Fig. 8 The training time T_{train} of the GEPSVM, LapSVM, LapTSVM and MPSVM on real-world datasets for the case of RBF kernel in the logarithmic scale, where m is the ratio of labeled data (Color figure online)

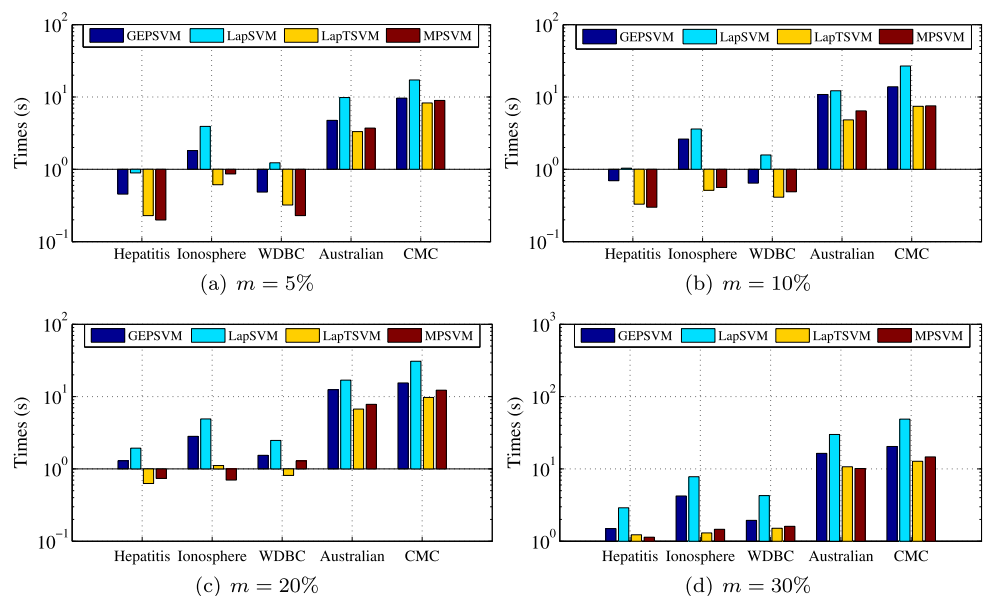


Fig. 9 The parameter searching time T_{para} of the GEPSVM, LapSVM, LapTSVM and MPSVM on real-world datasets for the case of RBF kernel in the logarithmic scale, where m is the ratio of labeled data (Color figure online)

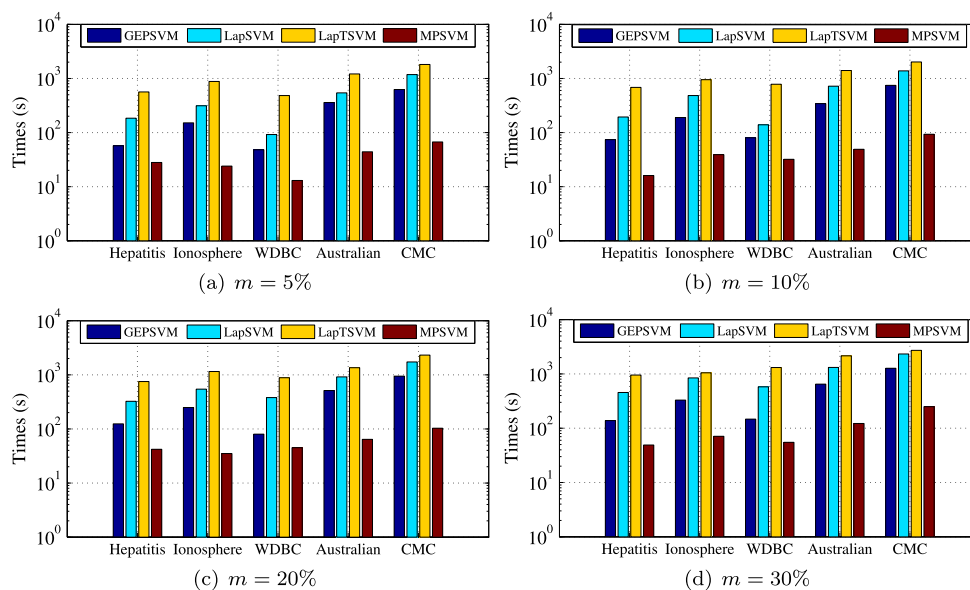


Fig. 10 An illustration of 10 subjects in the USPS database

USPS database consists of grayscale images of handwritten digits from ‘0’ to ‘9’, as shown in Fig. 10. The size of each image is 16×16 pixels with 256 gray levels. Similarly to [31], we choose four pairwise digits (200 images) on raw pixel features for our comparisons, and set up experiments in the following way. First, each dataset was divided into two subsets: 150 images for training and 50 images for testing. Then, we randomly labeled 40 images for the training set with m unlabeled images chosen from the remainder. Finally, we transformed them into semi-supervised tasks. Each experiment was repeated 10 times.

Figure 11 plots the learning results of each algorithm using a linear kernel, and includes the mean and deviation of testing accuracy for values of m from 20 to 100. As demonstrated in this figure, MPSVM generally has an obvious superiority over the other classifiers.

Overall, our MPSVM obtains significantly better classification accuracy than the other classifiers, but with remarkably lower learning time.

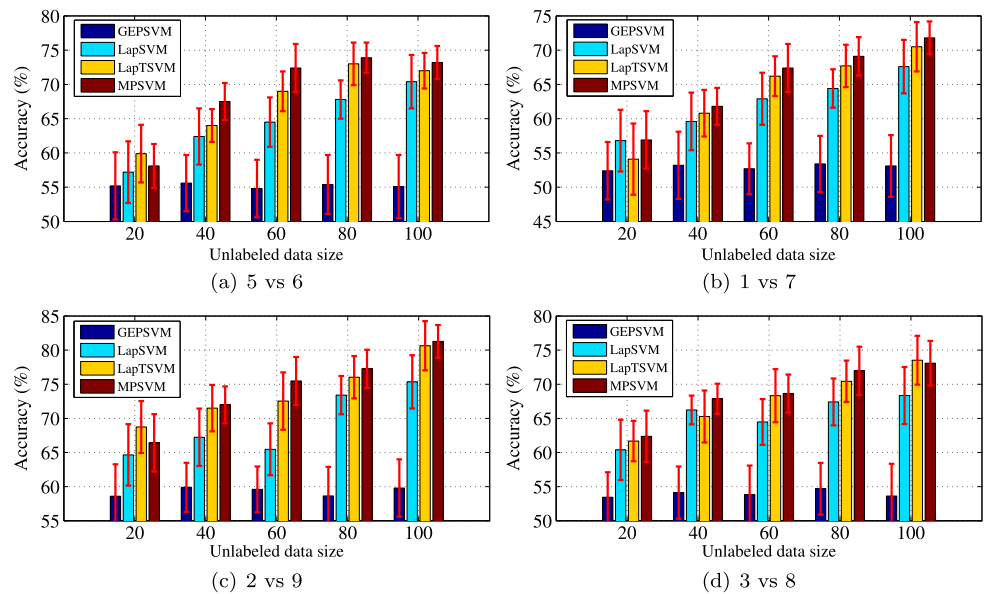
6 Conclusions

In this paper, we have proposed a novel MPSVM for binary semi-supervised classification. MPSVM incorporates both discriminant information and underlying geometric information to construct a more reasonable classifier. The optimal nonparallel proximal hyperplanes of MPSVM are obtained by solving a pair of standard eigenvalue problems. In addition, we designed an efficient PSO-based model selection approach, instead of a conventional grid search. We carried out a series of experiments to analyze our classifier against three state-of-the-art learning classifiers. The results demonstrate that MPSVM obtains significantly better performance than supervised GEPSVM, and achieves comparable or better performance than LapSVM and LapTSVM, with greater computational efficiency (including training time and parameter search time).

One of our future work is to construct the sparse L matrix in $\|f(\mathbf{X})\|_{\mathcal{M}}^2$ for the underlying manifold (distribution) representation. We also feel that extending our MPSVM to semi-supervised feature selection and multi-category classification would be interesting.

Acknowledgements The authors would like to thank the editors and the anonymous reviewers, whose invaluable comments helped improve the presentation of this paper substantially. This work is supported by the National Natural Science Foundation of China (11201426, 61203133, 11301485 and 61304125), the Zhejiang Provincial Natural Science Foundation of China (LQ12A01020, LQ13F030010) and the Science and Technology Foundation of Department of Education of Zhejiang Province (Y201225179).

Fig. 11 The test accuracy and standard deviation of GEPSVM, LapSVM, LapTSVM and MPSVM on USPS dataset for the case of linear kernel (Color figure online)



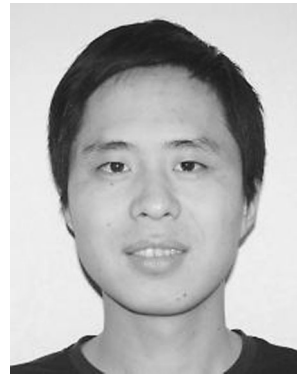
References

- Vapnik VN (1998) Statistical learning theory. Wiley, New York
- Burges CJC (1998) A tutorial on support vector machines for pattern recognition. *Data Min Knowl Discov* 2(2):121–167
- Deng N, Tian Y, Zhang C (2013) Support vector machines: theory, algorithms and extensions. CRC Press, Philadelphia
- Hao P, Chiang J, Lin Y (2009) A new maximal-margin spherical-structured multi-class support vector machine. *Appl Intell* 30(2):98–111
- Zhang HH, Ahn J, Lin XD, Park C (2006) Gene selection using support vector machines with non-convex penalty. *Bioinformatics* 22(1):88–95
- Lee L, Wan C, Rajkumar R, Isa D (2012) An enhanced support vector machine classification framework by using Euclidean distance function for text document categorization. *Appl Intell* 37(1):80–99
- Lee L, Rajkumar R, Isa D (2012) Automatic folder allocation system using Bayesian-support vector machines hybrid classification approach. *Appl Intell* 36(2):295–307
- Wang C, You W (2013) Boosting-SVM: effective learning with reduced data dimension. *Appl Intell* 39(3):465–474
- Mangasarian OL, Wild EW (2006) Multisurface proximal support vector machine classification via generalized eigenvalues. *IEEE Trans Pattern Anal Mach Intell* 28(1):69–74
- Shao Y, Deng N, Chen W, Zhen W (2013) Improved generalized eigenvalue proximal support vector machine. *IEEE Signal Process Lett* 20(3):213–216
- Ye Q, Zhao C, Zhang H, Ye N (2011) Distance difference and linear programming nonparallel plane classifier. *Expert Syst Appl* 38(8):9425–9433
- Jayadeva KR, Chandra S (2007) Twin support vector machines for pattern classification. *IEEE Trans Pattern Anal Mach Intell* 29(5):905–910
- Shao Y, Zhang C, Wang X, Deng N (2011) Improvements on twin support vector machines. *IEEE Trans Neural Netw* 22(6):962–968
- Peng X (2011) TPMSVM: a novel twin parametric-margin support vector machine for pattern recognition. *Pattern Recognit* 44(10–11):2678–2692
- Qi Z, Tian Y, Shi Y (2013) Structural twin support vector machine for classification. *Knowl-Based Syst* 43:74–81
- Shao Y, Deng N, Yang Z, Chen W, Wang Z (2012) Probabilistic outputs for twin support vector machines. *Knowl-Based Syst* 33:145–151
- Shao Y, Deng N, Yang Z (2012) Least squares recursive projection twin support vector machine for classification. *Pattern Recognit* 45(6):2299–2307
- Qi Z, Tian Y, Shi Y (2012) Twin support vector machine with universum data. *Neural Netw* 36:112–119
- Qi Z, Tian Y, Shi Y (2013) Robust twin support vector machine for pattern classification. *Pattern Recognit* 46(1):305–316
- Ding S, Yu J, Qi B, Huang H (2013) An overview on twin support vector machines. *Artif Intell Rev.* doi:10.1007/s10462-012-9336-0
- Chapelle O, Schölkopf B, Zien A (2010) Semi-supervised learning. MIT Press, Massachusetts
- Zhu X, Goldberg AB (2009) Introduction to semi-supervised learning. Morgan & Claypool, San Rafael
- Tur G, Hakkani D, Schapire RE (2005) Combining active and semi-supervised learning for spoken language understanding. *Speech Commun* 45(2):171–186
- Guzella TS, Caminhas WM (2009) A review of machine learning approaches to spam filtering. *Expert Syst Appl* 36(7):10206–10222
- Zhang T, Liu S, Xu C, Lu H (2011) Boosted multi-class semi-supervised learning for human action recognition. *Pattern Recognit* 44(10–11):2334–2342
- Nguyen T, Ho T (2012) Detecting disease genes based on semi-supervised learning and protein protein interaction networks. *Artif Intell Med* 54(1):63–71
- Soares RGF, Chen H, Yao X (2012) Semisupervised classification with cluster regularization. *IEEE Trans Neural Netw Learn Syst* 23(11):1779–1792
- Fan M, Gu N, Qiao H, Zhang B (2011) Sparse regularization for semi-supervised classification. *Pattern Recognit* 44(8):1777–1784
- Belkin M, Niyogi P, Sindhvani V (2006) Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. *J Mach Learn Res* 7:2399–2434
- Melacci S, Belkin M (2011) Laplacian support vector machines trained in the primal. *J Mach Learn Res* 12:1149–1184
- Qi Z, Tian Y, Shi Y (2012) Laplacian twin support vector machine for semi-supervised classification. *Neural Netw* 35:46–53

32. Chen W, Shao Y, Ye Y (2013) Improving Lap-TSVM with successive overrelaxation and differential evolution. *Proc Comput Sci* 17:33–40
33. Chen W, Shao Y, Hong N (2013) Laplacian smooth twin support vector machine for semi-supervised classification. *Int J Mach Learn Res Cybern*. doi:10.1007/s13042-013-0183-3
34. Tikhonov AN, Arsenin VY (1979) *Methods for solving ill-posed problems*. Nauka, Moscow
35. Parlett B (1998) *The symmetric eigenvalue problem*. SIAM, Philadelphia
36. Lin SW, Ying KC, Chen SC, Lee ZJ (2008) Particle swarm optimization for parameter determination and feature selection of support vector machines. *Expert Syst Appl* 35(4):1817–1824
37. Shao Y, Wang Z, Chen W, Deng N (2013) Least squares twin parametric-margin support vector machine for classification. *Appl Intell* 39(3):451–464
38. Huang CL, Dun JF (2008) A distributed pso-svm hybrid system with feature selection and parameter optimization. *Appl Soft Comput* 8(4):1381–1391
39. Das S, Suganthan PN (2011) Differential evolution: a survey of the state-of-the-art. *IEEE Trans Evol Comput* 15(1):4–31
40. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: *IEEE international conference on neural networks*, vol 4, pp 1942–1948
41. Poli R, Kennedy J, Blackwell T (2007) Particle swarm optimization. *Swarm Intell* 1(1):33–57
42. Gan H, Sang N, Huang R, Tong X, Dan Z (2013) Using clustering analysis to improve semi-supervised classification. *Neurocomputing* 101:290–298
43. Yang Z, Fang K, Kotz S (2007) On the student's t-distribution and the t-statistic. *J Multivar Anal* 98(6):1293–1304



Wei-Jie Chen received his B.S. degree and Ph.D. degree both in College of Information at Zhejiang University of Technology, China, in 2006 and 2011. He is now a lecturer at the Zhijiang College, Zhejiang University of Technology. His research interests include pattern recognition, intelligence computation and manifold learning. He has published over 20 refereed papers.



Yuan-Hai Shao received his B.S. degree in College of Mathematics from Jilin University, and received Ph.D. degree in College of Science from China Agricultural University, China, in 2006 and 2011, respectively. Currently, he is a lecturer at the Zhijiang College, Zhejiang University of Technology. His research interests include optimization methods, machine learning and data mining. He has published over 30 refereed papers.



Deng-Ke Xu received his Ph.D. degree in College of Applied Sciences at Beijing University of Technology, China, in 2013. He is now a lecturer at the Department of Statistics, Zhejiang Agriculture and Forest University. His main research interests include variable selection, joint mean and variance models and Bayesian analysis.



Yong-Feng Fu received her Ph.D. degree in Department of Control Science and Engineering from Zhejiang University, China, in 2007. Currently, she is an associate professor in Zhijiang College, Zhejiang University of Technology. Her research interests include soft sensor modeling methods and their applications in industrial process, modeling and optimization of complicated industrial process.