

Manifold proximal support vector machine with mixed-norm for semi-supervised classification

Zhiqiang Zhang · Ling Zhen · Naiyang Deng · Junyan Tan

Received: 11 April 2014 / Accepted: 14 September 2014 / Published online: 1 October 2014
© The Natural Computing Applications Forum 2014

Abstract Since labeling all the samples by the user is time-consuming and fastidious, we often obtain a large amount of unlabeled examples and only a small number of labeled examples in classification. In this context, the classification is called semi-supervised learning. In this paper, we propose a novel semi-supervised learning methodology, named Laplacian mixed-norm proximal support vector machine Lap-MNPSVM for short. In the optimization problem of Lap-MNPSVM, the information from the unlabeled examples is used in a form of Laplace regularization, and l_p norm ($0 < p < 1$) regularizer is introduced to standard proximal support vector machine to control sparsity and the feature selection. To solve the nonconvex optimization problem in Lap-MNPSVM, an efficient algorithm is proposed by solving a series systems of linear equations, and the lower bounds of the solution are established, which are extremely helpful for feature selection. Experiments carried out on synthetic datasets and the real-world datasets show the feasibility and effectiveness of the proposed method.

Keywords Semi-supervised classification · Manifold regularization · Mixed-norm · Sparsity · Proximal support vector machine

Z. Zhang
School of Mechanical Engineering, Beijing Institute of Technology, Beijing 100081, China
e-mail: sunwithmoon@bit.edu.cn

L. Zhen · N. Deng · J. Tan (✉)
College of Science, China Agricultural University, Beijing 100083, China
e-mail: tanjunyan0@126.com

1 Introduction

In many real situations, there are plentiful unlabeled training examples since the acquisition of labels is time-consuming and fastidious. In such situations, semi-supervised learning tries to utilize the unlabeled examples to improve learning performance, especially when there are limited labeled training examples. During the past decade, semi-supervised learning has received significant attention and a lot of approaches have been developed [1, 2–10].

Among the many semi-supervised learning methods, manifold regularization (MR) is one of the most interesting frameworks [11–14]. The MR introduces a meaningful regularization term to capture the geometric information from the data and makes the smoothness of the classifier along the intrinsic manifold. Following the MR framework, Belkin et al. proposed a Laplacian support vector machine (Lap-SVM) [12], which is able to handle both the transductive and inductive cases. Qi et al. [28] proposed Laplacian twin support vector machine (Lap-TSVM), which constructs a nonparallel classifier for semi-supervised learning.

The semi-supervised learning methods proposed in this paper are closely related with the proximal support vector machine (PSVM) [15] and the p -norm ($0 < p < 1$) support vector machine (p -norm SVM) [21–24] for supervised classification problem. Different from the standard 2-norm SVM, PSVM generates the linear classifier based on proximity to one of two parallel planes that are pushed as far apart as possible. It only requires solving a simple nonsingular system of linear equations (LEs), while the standard 2-norm support vector machine classifier requires a more costly solution of a quadratic program. The p -norm SVM comes from the good contribution on the p -norm ($p \in [0, 1]$) in the optimization communities [17–25],

where the 2-norm penalty in the standard linear SVM is replaced by the p -norm ($p \in (0, 1)$) penalty.

In this paper, we introduced two extra regularizers into the optimization problem of PSVM. One is the manifold regularizer, which captures the geometric structure of the unlabeled and labeled examples, the other is the p -norm, which can control the sparsity of hyperplane and realize feature selection. The proposed method is called Laplacian mixed-norm proximal support vector machine (Lap-MNPSVM). Our Lap-MNPSVM can realize classification and feature selection simultaneously. Unfortunately, the optimization problem of our Lap-MNPSVM is neither convex nor differentiable; it is therefore difficult to be solved directly. We propose an algorithm to find its approximate solution via solving a series systems of LEs. And the lower bounds for the absolute value of nonzero components in every local optimal solution are established, which are extremely helpful to eliminate the zero components in any numerical solution. The numerical experimental results have shown that our Lap-MNPSVM is more effective than some popular semi-supervised learning methods such as transductive SVM (TSVM) [27], Laplacian Regularized Least Square Classifier (Lap-RLS) [12], Laplacian SVM (Lap-SVM) [12] and Laplacian Twin SVM (Lap-TSVM) [28].

This paper is organized as follows. In Sect. 2, we briefly introduce some works related to this paper first and then describe our Lap-PPSVM in detail including solving and analyzing the involved optimization problem in it. In Sect. 3, numerical experiments are given to demonstrate the effectiveness of our method. We conclude this paper in Sect. 4.

2 Methods

In this section, we first remind proximal support vector machine (PSVM) and the semi-supervised manifold regularization; then, we carry out our Lap-MNPSVM.

We describe our notation firstly. All vectors are column vectors unless transposed to a row vector by a super script \top . For a vector x in R^n , $[x]_i (i = 1, 2, \dots, n)$ denotes the i th component of x . $|x|$ denotes a vector in R^n of absolute value of the components of x . $\|x\|_p$ denotes the value $(|x_1|^p + \dots + |x_n|^p)^{\frac{1}{p}}$. Strictly speaking, when $0 < p < 1$ $\|x\|_p$ is not a norm in general sense.¹ But, we still follow the term p -norm because of simplicity and the consistence with the other references such as [17–25]. $\|x\|_0$ is the number of nonzero components of x . For two vectors $x =$

$([x]_1, \dots, [x]_n)^\top \in R^n$ and $y = ([y]_1, \dots, [y]_n)^\top \in R^n$, $(x \cdot y)$ denotes the inner product of x and y , $x \otimes y$ generates a new vector with the i -th element $[x]_i [y]_i, i = 1, 2, \dots, n$.

Consider the semi-supervised classification problem with the training set T

$$T = \{(x_1, y_1), \dots, (x_l, y_l), x_{l+1}, \dots, x_{l+u}\}, \tag{1}$$

where $x_j \in R^n, j = 1, 2, \dots, l + u$ and $y_j \in \{1, -1\} (j = 1, \dots, l)$. Denote the inputs of all examples $X = \{x_i\}_{i=1}^{l+u} \in R^{(l+u) \times n}$ and each row $X_i \in R^n$ is the input of the i -th example. Suppose the inputs of labeled examples denoted by $X_l = \{x_i\}_{i=1}^l \in R^{l \times n}$, the outputs of the labeled examples denoted by $Y_l = \{y_i\}_{i=1}^l \in R^{l \times 1}$. Our goal is to construct a classifier utilizing both labeled and unlabeled examples, which can realize the feature selection and give a better generalization performance.

2.1 Proximal support vector machine (PSVM)

For supervised classification problem, instead of the standard support vector machine (SVM) that classifies the examples by assigning them to one of two disjoint half spaces in input or feature space, PSVM assigns examples to the closer one of two parallel super planes. Its optimization problem is as follows,

$$\min_{w, b, \xi} \frac{1}{2} (\|w\|_2^2 + b^2) + \frac{C}{2} \sum_{i=1}^l \xi_i^2, \tag{2}$$

$$\text{s.t. } y_i((w \cdot x_i) + b) = 1 - \xi_i, i = 1, \dots, l. \tag{3}$$

The first term in (2) is the regularizer, optimizing it can maximize the margin between two boundary hyperplanes $((w \cdot x) + b = 1$ and $(w \cdot x) + b = -1)$ and avoid over-fitting. Minimizing the second term is to minimize the empirical risk. It is clear that PSVM requires only solving a nonsingular system of LEs.

2.2 Semi-supervised manifold regularization

Recently, manifold learning techniques [11, 13, 16] have attracted much attention as they can preserve some geometric information of the data. Particularly, for semi-supervised learning (SSL), Belkin et al. [12] introduced a meaningful regularization term:

$$\|f\|_M^2 = \sum_{ij=1}^{l+u} w_{ij} (f(x_i) - f(x_j))^2 = f^\top L f, \tag{4}$$

where $f = (f(x_1), \dots, f(x_{l+u}))^\top$ represents the decision function values over all the training examples, $W = (w_{ij}) \in R^{(l+u) \times (l+u)}$, w_{ij} is the edge-weight pre-defined for a pair of points (x_i, x_j) , and D is the diagonal matrix given by

¹ $\|x\|_p (0 < p < 1)$ is a quasi-norm, which satisfies the norm axioms except the triangle inequality.

$D_{ii} = \sum_{j=1}^{l+u} w_{ij}, L = D - W$. It is easy to see that the MR term contains all the information from the labeled and unlabeled examples and is suitable for SSL.

2.3 Laplacian mixed-norm proximal SVM (Lap-MNPSVM)

2.3.1 Linear Lap-MNPSVM

Now, we are in a position to present our novel algorithm—the Laplacian p-norm proximal support vector machine (Lap-MNPSVM) by modifying the above problems (2–3). We hope our algorithm can automatically realize feature selection and classification simultaneously. To realize the former, we replace the $\|w\|_2^2$ in the objective function (2) by $\frac{1}{2}\|w\|_2^2 + \frac{1}{2}\|w\|_p^p$ ($0 < p < 1$). To realize the latter, we add the extra regularization term (4) to the objective function (2), making our Lap-MNPSVM is suitable for SSL. This leads to the optimization problem:

$$\min_{w,b,\xi} \frac{1}{2} \left(\|w\|_2^2 + b^2 \right) + \frac{1}{2} \|w\|_p^p + \frac{C}{2} \sum_{i=1}^l \xi_i^2 + \frac{\lambda}{2} \sum_{ij=1}^{l+u} w_{ij} (f(x_i) - f(x_j))^2, \tag{5}$$

$$\text{s.t. } y_i((w \cdot x_i) + b) = 1 - \xi_i, \quad i = 1, \dots, l, \tag{6}$$

where $w_{ij} = \exp(-\frac{\|x_i - x_j\|_2}{\sigma})$, σ is the parameter that can be adjusted. Substituting the equality constraints and $f(x_i) = (w \cdot x_i) + b$ into the objective function, we get the simplified version:

$$\min_{w,b} \frac{1}{2} \left(\|w\|_2^2 + b^2 \right) + \frac{1}{2} \|w\|_p^p + \frac{C}{2} \sum_{i=1}^l [y_i - ((w \cdot x_i) + b)]^2 + \frac{\lambda}{2} \sum_{ij=1}^{l+u} w_{ij} [(w \cdot x_i) + b - ((w \cdot x_j) + b)]^2, \tag{7}$$

where $C(C > 0), \lambda(0 \leq \lambda \leq 1), \mu > 0$ and $p(0 < p < 1)$ are parameters.

We now give the geometric interpretation of problem (7). The first term is the regularizer, optimizing it can maximize the margin between two boundary hyperplanes. The second term is also the regularizer that can control the sparsity² of the final classification hyperplane. The third term minimizes the squared sum of errors, which makes the examples to be classified as correct as possible. The last term, (MR) term aims at to exploit the geometric information inside all the examples, and enforces $f(x) = (w \cdot x) + b$ smoothness along the intrinsic manifold M. In

² Sparsity is here defined as the number of nonzero components in the normal vector w . This means that more zero components in w , more sparse the hyperplane.

addition, regularization parameter λ is introduced to balance the relative significance between the empirical risk term and MR term.

Note that, it is rather difficult to find the global solution of problem (7) because its objective function is neither convex nor differentiable. For the issue of nondifferentiable, we approximate $\|w\|_p^p = \sum_{i=1}^n |[w]_i|^p$ by $\sum_{i=1}^n (|[w]_i| + \varepsilon)^p$ and get the following problem

$$\min_{w,b} F_p(w,b) = \frac{1}{2} \left(\|w\|_2^2 + b^2 + \sum_{i=1}^n (|[w]_i| + \varepsilon)^p \right) + \frac{C}{2} \sum_{i=1}^l [y_i - ((w \cdot x_i) + b)]^2 + \frac{\lambda}{2} \sum_{i=1}^{l+u} \sum_{j=1}^{l+u} w_{ij} [(w \cdot x_i) + b - ((w \cdot x_j) + b)]^2 \tag{8}$$

here, $\varepsilon > 0$ is a very small number. But this term is still concave due to $0 < p < 1$. To solve this issue, the convex term $\frac{1}{2} \|\beta \otimes w\|_2^2$ is used to approximate the concave term $\sum_{i=1}^n (|[w]_i| + \varepsilon)^p$, β is adjustable to fit the approximation. Thus, we get the following convex quadratic program that approximated problem (7):

$$\min_{w,b} \frac{1}{2} \left(\|w\|_2^2 + b^2 \right) + \frac{1}{2} \|\beta \otimes w\|_2^2 + \frac{C}{2} \sum_{i=1}^l [y_i - ((w \cdot x_i) + b)]^2 + \frac{\lambda}{2} \sum_{ij=1}^{l+u} w_{ij} [(w \cdot x_i) + b - ((w \cdot x_j) + b)]^2. \tag{9}$$

In order to get better approximation, we adjust β successively. This is an iterative process as follows:

Given an initial $\beta^{(0)} = (\beta_1^{(0)}, \dots, \beta_n^{(0)})^\top$, suppose that the current $(w^{(k)}, b^{(k)})$ is estimated by solving (9) with $\beta = \beta^{(0)}$. Set $(w^{(k+1)}, b^{(k+1)})$ as the solution to the following weighted optimization problem:

$$\min_{w,b} F_2(w,b) = \frac{1}{2} \left(\|w\|_2^2 + b^2 \right) + \frac{1}{2} \left(\|\beta^{(k+1)} \otimes w\|_2^2 \right) + \frac{C}{2} \sum_{i=1}^l [y_i - ((w \cdot x_i) + b)]^2 + \frac{\lambda}{2} \sum_{ij=1}^{l+u} w_{ij} [(w \cdot x_i) + b - ((w \cdot x_j) + b)]^2, \tag{10}$$

where $\beta^{(k+1)} = (\beta_1^{(k+1)}, \dots, \beta_n^{(k+1)})^\top$ is the weight vector and is required to satisfy:

$$p\left(\left|[w^{(k)}]_i\right| + \varepsilon\right)^{p-1} \text{sign}\left([w^{(k)}]_i\right) = \left[\beta_i^{(k+1)}\right]^2 [w^{(k)}]_i. \tag{11}$$

Equation (11) means that the objective function of (8) has the same steepest descent direction as the objective function of problem (10) at the current $(w^{(k)}, b^{(k)})$, i.e.,

$$\nabla F_p(w^{(k)}, b^{(k)}) = \nabla F_2(w^{(k)}, b^{(k)}), \tag{12}$$

So a reasonable choice is

$$\beta_i^{(k+1)} = \sqrt{p\left(\left|[w^{(k)}]_i\right| + \varepsilon\right)^{p-2}}. \tag{13}$$

Based on the above discussion, we can get the approximated solution $(w^{(k+1)}, b^{(k+1)})$ of problem (7) by solving (10). Note that problem (10) is a unconstrained quadratic programming. Its KKT condition leads to solving the following LEs:

$$(B + C\bar{X}_l^\top \bar{X}_l + \lambda \bar{X}^\top L \bar{X})\tilde{w} = \bar{X}_l^\top Y_l, \tag{14}$$

where

$$\bar{X} = \begin{pmatrix} x_1^\top & 1 \\ x_2^\top & 1 \\ \vdots & \vdots \\ x_{l+u}^\top & 1 \end{pmatrix}_{(l+u) \times (n+1)}, \quad \bar{X}_l = \begin{pmatrix} x_1^\top & 1 \\ x_2^\top & 1 \\ \vdots & \vdots \\ x_l^\top & 1 \end{pmatrix}_{l \times (n+1)}, \quad \tilde{w} = [w^\top b]^\top,$$

and $B = \text{diag}(1 + [\beta_1^{(k+1)}]^2, \dots, 1 + [\beta_n^{(k+1)}]^2, 1)$.

The above LEs can be solved by the powerful conjugate gradient (CG) algorithm [26] effectively, which is described as follows: Suppose the linear system is represented as

$$Ax = b, \tag{15}$$

where x is an unknown solution, A is a symmetric and positive definite matrix, and b is a vector. The whole procedure is shown by Algorithm 1.

Algorithm 1 The conjugate gradient algorithm for problem (15)

Input:
 The matrix A and the vector b .
 The prescribed convergence constant ε_0 .
 The approximate initial solution x_0 .
 1: Initialize the residual vector $r_0 = b - Ax_0$ and the search direction $p_0 = r_0$.
 2: **Do While** $\|r_{k+1}\| < \varepsilon_0$
 3: Set the step scalar $a_k = r_k^\top r_k / p_k^\top A p_k$
 4: Update the new solution $x_{k+1} = x_k + a_k p_k$
 5: Renew the residual vector $r_{k+1} = r_k - a_k A p_k$
 6: Set the step scalar $b_k = r_k^\top r_k / r_k^\top r_k$
 7: Modify the next search direction $p_{k+1} = r_{k+1} + b_k p_k$
 8: Increase iterator $k = k + 1$
 9: **End While**
Output:
 The optimal solution for the problem (15).

So in this way it can be expected to find a solution of (7). Now, there arises another issue: Since we can only

get the approximate local solution of problem (7) by iteratively solving weighted-biased SVM (14) using algorithm 1, how to identify the nonzero components in the solution? To solve this issue, we prove the following theorem, which can be used to identify nonzero components in any local optimal solutions from an approximate local optimal solution.

Theorem 1 For any local optimal solution (w^*, b^*) to the problem (7), we have $[w^*]_i = 0$ if $[w^*]_i \in (-L_i, L_i)$, where $L_i = \left[\frac{p}{2|C \sum_{j=1}^l [y_j]_i (y_j - b^*)|}\right]^{\frac{1}{p-1}}$, $(i = 1, 2, \dots, n)$.

Proof Suppose $\|w^*\|_0 = k$. Without loss of generality, let $w^* = ([w^*]_1, [w^*]_2, \dots, [w^*]_k, 0, 0, \dots, 0)^T$ and $z^* = ([w^*]_1, [w^*]_2, \dots, [w^*]_k)^T$. Construct the new training set

$$\tilde{T} = \{(\tilde{x}_1, y_1), \dots, (\tilde{x}_l, y_l), \tilde{x}_{l+1}, \dots, \tilde{x}_{l+u}\},$$

where $\tilde{x}_i = ([x_i]_1, [x_i]_2, \dots, [x_i]_k)^T$. We consider the following optimization problem

$$\begin{aligned} \min_{z, b} F_p(z, b) &= \frac{1}{2} \left(\|z\|_2^2 + \|z\|_p^p + b^2 \right) \\ &+ \frac{C}{2} \sum_{i=1}^l [y_i - ((z \cdot \tilde{x}_i) + b)]^2 \\ &+ \frac{\lambda}{2} \sum_{ij=1}^{l+u} w_{ij} [((z \cdot \tilde{x}_i) + b) - ((z \cdot \tilde{x}_j) + b)]^2. \end{aligned} \tag{16}$$

It is easy to know that (z^*, b^*) is a local optimal solution of (16), according to the KKT condition, we have

$$\begin{aligned} I_n z^* + \tilde{X}_l^\top \tilde{X}_l z^* + \frac{1}{2} p |z^*|^{p-1} \text{sign}(z^*) \\ + \tilde{X}^\top (D - W) \tilde{X} z^* = C \tilde{X}_l^\top (Y_l - e b^*) \end{aligned} \tag{17}$$

where I_n is a n dimensional identity matrix and

$$\begin{aligned} \tilde{X} &= \begin{pmatrix} \tilde{x}_1^\top \\ \tilde{x}_2^\top \\ \vdots \\ \tilde{x}_{l+u}^\top \end{pmatrix}_{(l+u) \times k}, \quad \tilde{X}_l = \begin{pmatrix} \tilde{x}_1^\top \\ \tilde{x}_2^\top \\ \vdots \\ \tilde{x}_l^\top \end{pmatrix}_{l \times k}, \\ e &= \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}_{l \times 1}. \end{aligned}$$

By (17), we have

$$\begin{aligned} \frac{1}{2}p|z^*|^{p-1} &\leq |L_n z^* + \tilde{X}_l^T \tilde{X}_l z^* + \frac{1}{2}p|z^*|^{p-1} \text{sign}(z^*) \\ &\quad + \tilde{X}^T (D - W) \tilde{X} z^* \\ &= |C \tilde{X}_l^T (Y_l - e b^*)|. \end{aligned} \tag{18}$$

So,

$$\frac{1}{2}p|[z^*]_i|^{p-1} \leq |C \sum_{j=1}^l [x_j]_i (y_j - b^*)|, \tag{19}$$

which is equivalent to $|[z^*]_i| \geq \left[\frac{p}{2|C \sum_{j=1}^l [x_j]_i (y_j - b^*)|} \right]^{\frac{1}{1-p}}$. It

means that for any local optimal solution (w^*, b^*) of (7), we have $[w^*]_i \in (-L_i, L_i) \Rightarrow [w^*]_i = 0, i = 1, 2, \dots, n$.

Based on the above discussion, our novel algorithm is established as follows:

Algorithm 2 Laplacian p-norm proximal support vector machine (Lap-PPSVM)

Input:

- The training set (1), parameters $C(C > 0)$, $p(0 < p < 1)$ and $\lambda > 0$.
- A specified maximum number of iterations K_{max} and a very small $\varepsilon_1 > 0$.
- 1: construct the optimization problem (7);
- 2: Start with a random $\beta^{(0)}$ and let $k = 1$, using Algorithm 1 solving problem (14), get the solution $(w^{(k)}, b^{(k)})$ and update $\beta^{(k+1)}$ according to (13)
- 3: If $\|\beta^{(k+1)} - \beta^{(k)}\| < \varepsilon_1$ or k attains K_{max} , terminate and set the optimal solution of (7) $(w^*, b^*) = (w^{(k)}, b^{(k)})$; Otherwise, set $k = k + 1$ and update the weights for each $i = 1, 2, \dots, n$ according to (13) and go to step 2

Output:

The optimal solution (w^*, b^*) , the feature index set: $F' = \{i|[w^*]_i > L_i, i = 1, \dots, n\}$; The decision function $f(x) = \text{sgn}((\tilde{w}^* \cdot \tilde{x}) + b^*)$, where \tilde{w}^* are composed by the components in the F' of w^* and the components of \tilde{x} are also corresponding to components in the feature set F' of w^* .

2.3.2 Nonlinear Lap-MNPSVM

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

$$K(x, X^T)u + b = 0, \tag{20}$$

where K is an chosen kernel function: $K(x_i, x_j) = (\varphi(x_i) \cdot \varphi(x_j))$. The optimization problems for our nonlinear l_p LSTSVM can be expressed as

$$\begin{aligned} \min_{u,b} \frac{1}{2} \left(\|u\|_2^2 + \|u\|_p^p + b^2 \right) &+ \frac{C}{2} \sum_{i=1}^l (K(x_i, X^T)u + b - y_i)^2 \\ &+ \frac{\lambda}{2} \sum_{i,j=1}^{l+u} [(K(x_i, X^T)u - K(x_j, X^T)u)w_{ij}]^2. \end{aligned} \tag{21}$$

Now, we rewrite problem (21) into the following equivalent form:

$$\min_{u,b} \frac{1}{2} \left(\|u\|_2^2 + \|u\|_p^p + b^2 \right) + \frac{C}{2} \|\tilde{K}_l \tilde{u} - Y_l\|_2^2 + \frac{\lambda}{2} \tilde{u}^T \tilde{K} L \tilde{K} \tilde{u}, \tag{22}$$

where

$$\tilde{u} = \begin{bmatrix} u \\ b \end{bmatrix}, \tilde{K}_l = [K(X_l, X^T) \quad e], \tilde{K} = [K(X, X^T) \quad e].$$

With an entirely similar process to the linear case, problems (22) can be solved iteratively. At the k -th iteration, suppose that the current (u, b) is estimated by $(u^{(k)}, b^{(k)})$. Set $(u^{(k+1)}, b^{(k+1)})$ as the solution to the following weighted QPP:

$$\begin{aligned} \min_{u,b} \frac{1}{2} \left(\|u\|_2^2 + \|\beta^{(k+1)} \otimes u\|_2^2 + b^2 \right) &+ \frac{C}{2} \|\tilde{K}_l \tilde{u} - Y_l\|_2^2 \\ &+ \frac{\lambda}{2} \tilde{u}^T \tilde{K} L \tilde{K} \tilde{u}, \end{aligned} \tag{23}$$

where

$$[\beta^{(k+1)}]_i = \sqrt{p \left(|u_1^{(k)}|_i + \varepsilon \right)^{p-2}}. \tag{24}$$

Note that problems (23) is an unconstrained QPP, according to the KKT conditions, it is equivalent to the following LEs:

$$(\mathcal{B}_1 + C \tilde{K}_l^T \tilde{K}_l + \lambda \tilde{K} L \tilde{K}) \tilde{u} = \tilde{K}_l^T Y_l, \tag{25}$$

where $\mathcal{B}_1 = \text{diag}(1 + [\beta^{(k+1)}]_1^2, \dots, 1 + [\beta^{(k+1)}]_{l+u}^2, 1)$.

Then, we can use Algorithm 1 to solve the LEs (25). The optimal solutions (u, b) can be obtained by almost the same iterative progress as algorithm 2, and a new data $x \in R^n$ are assigned to class $i (i = +1 \text{ or } -1)$, depending on

$$f(x) = \text{sgn}K(x, X^T)u + b.$$

3 Experiment and results

To evaluate the performance of our Lap-MNPSVM, we investigate its classification accuracy and feature selection on synthetic dataset and some real-world datasets. We focus on the comparison between Lap-MNPSVM and several state-of-the-art semi-supervised classifiers, including TSVM, Lap-RLS, Lap-SVM and Lap-TSVM:

- TSVM [27]: Transductive SVM. It adopts the cluster assumption and attempts to seek a low-density region to separate classes (guided by the maximum margin principle), avoiding the boundary passing through the high-density region.
- Lap-RLS [12]: Laplacian Regularized Least Square Classifier. It adopts the manifold assumption and solves the optimization problem with the squared loss function (an extension of RLS [30] for SSL).
- Lap-SVM [12]: Laplacian SVM. It 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 (an extension of SVM for SSL).

- Lap-TSVM [28]: Laplacian Twin SVM. It also adopts the manifold assumption and exploits the geometric information embedded in the training data to construct a nonparallel hyperplane classifier (an extension of TWSVM for SSL).

Our algorithm code is written in MATLAB 2010 on a PC with an Intel Core I5 processor with 4 GB RAM. With regard to parameter selection, we employ the tenfold cross-validation technique on the training set. Parameters C, λ and σ in MR term are all selected from the set $\{2^i | i = -6, \dots, 6\}$, p is selected from the set $\{0.1, 0.2, \dots, 0.9\}$.

3.1 Results

3.1.1 Comparison on UCI datasets

It is well known that the scale of the labeled examples is important to the semi-supervised learning. To investigate the impact of the ratio of labeled data on the performance of Lap-MNPSVM, each classifier is applied on several real-world datasets from the UCI machine learning repository, summarized in Table 1. These datasets represent a wide range of fields (include pathology, vehicle engineering, biological information and finance), sizes (from 155 to 1,437) and features (from 6 to 34). All datasets are normalized such that the features scale in $[-1, 1]$ before training. Similar to [2], our experiments are set up in the following way. First, each dataset is divided into two subsets: 70 % for training and 30 % for testing. Then, we randomly label m of the training set and use the remainder as unlabeled examples, where m is the ratio of labeled examples. Finally, we transform them into semi-supervised tasks. Each experiment is repeated 10 times.

Accuracy (A_{cc}) is utilized to evaluate the performance of classification and is defined as follows. Accuracy denotes the percentage of both positive points and negative points correctly predicted and is defined as follows:

Table 1 Summary of UCI datasets

Datasets	Size	Features
Hepatitis	155	19
Hearts	270	14
Inosphere	351	34
WDBC	569	30
Australian	690	14
BUPA	345	6
Housevotes	435	16
CMC	1,473	9
German	1,000	20

Table 2 Mean and standard deviation (%) of test accuracy and the number of selected features at 10 % of labeled examples

Dataset	TSVM Acc Std feature	Lap- RLS Acc Std feature	Lap- SVM Acc Std feature	Lap- TSVM Acc Std feature	Lap- MNPSVM Acc Std feature
Hepatitis	72.64	73.19	74.92	74.11	79.56
	5.83	6.64	5.86	5.3522	6.36
	19	19	19	19	9.2
Hearts	70.09	65.53	67.87	70.76	72.1
	5.72	6.33	7.24	5.96	5.96
	14	14	14	14	12.4
Inosphere	68.24	72.18	69.54	72.95	78.46
	4.04	7.82	5.63	6.24	7.8
	34	34	34	34	21.4
WDBC	86.32	88.73	84.14	86.38	92.35
	6.55	7.14	5.92	6.62	3.19
	30	30	30	30	24.6
Australian	64.49	64.13	65.21	66.61	80.39
	5.82	6.44	7.53	7.04	3.67
	14	14	14	14	10.8
BUPA	63.72	66.18	65.09	65.87	54.56
	5.73	6.14	6.93	6.28	7.41
	6	6	6	6	6
Housevotes	87.14	84.69	89.17	87.53	94.15
	7.39	5.48	5.45	6.39	0.88
	16	16	16	16	13
CMC	59.72	59.27	58.72	61.39	77.55
	7.31	6.83	9.35	5.09	0.1
	9	9	9	9	9
German	57.71	58.24	59.77	60.56	71.73
	6.81	8.24	6.89	7.73	1.36
	20	20	20	20	16.4
Ave.mean	70.01	70.29	70.49	71.79	77.87
Ave.std	6.13	6.78	6.75	6.3	4.08

Ave.mean and Ave.std denote the average mean and standard deviation accuracy of each algorithm over all datasets

$$A_{cc} = \frac{TP + TN}{TP + FP + TN + FN},$$

where TP, TN, FP and FN denote the number of true positives, true negatives, false positives and false negatives, respectively.

Tables 2, 3, 4 list the learning results of each classifier and include the mean and deviation of test accuracy for various m from 10 to 30 %. We have highlighted the best performance. The results reveal that increasing the ratio of labeled examples generally improves the classification performance for almost all classifiers.

Now, we focus on the performance of our Lap-MNPSVM. The results in Tables 2, 3, 4 show that Lap-

Table 3 Mean and standard deviation (%) of test accuracy and the number of selected features at 20 % of labeled examples

Dataset	TSVM Acc Std feature	Lap- RLS Acc Std feature	Lap- SVM Acc Std feature	Lap- TSVM Acc Std feature	Lap- MNPSVM ACC Std feature
Hepatitis	74.47	73.42	74.92	76.87	81.78
	6.96	7.24	5.86	7.39	2.9
	19	19	19	19	18.8
Hearts	69.23	68.17	73.19	72.95	77.28
	6.49	5.27	7.79	6.49	5.56
	14	14	14	14	12
Inosphere	69.12	73.09	72.98	74.25	83.65
	7.54	7.34	5.68	6.65	3.6
	34	34	34	34	31.6
WDBC	92.93	90.20	91.84	93.53	92.94
	6.74	6.75	6.31	7.17	3.77
	30	30	30	30	28.4
Australian	66.19	64.38	67.45	65.96	85.53
	6.73	7.13	8.10	7.04	3.57
	14	14	14	14	13.2
BUPA	65.17	63.11	65.91	67.85	62.33
	8.09	8.28	8.15	6.59	3.39
	6	6	6	6	6
Housevotes	89.34	86.58	90.01	91.67	95.54
	7.26	7.35	6.59	6.52	1.38
	16	16	16	16	15.2
CMC	61.05	60.71	62.73	63.41	77.55
	6.48	8.03	7.31	5.91	0.1
	9	9	9	9	9
German	59.34	60.40	62.18	61.70	72.13
	8.17	8.31	7.20	6.83	2.06
	20	20	20	20	20
Ave.mean	71.87	71.12	73.46	74.24	80.97
Ave.std	7.16	7.30	7.00	6.73	2.93

Ave.mean and Ave.std denote the average mean and standard deviation accuracy of each algorithm over all datasets

MNPSVM is better than other methods on almost all datasets. The accuracy is improved in varying degrees using Lap-MNPSVM. In particular, on the dataset ‘Australian,’ ‘CMC’ and ‘German,’ there are over 10 % improvements using Lap-MNPSVM. Only on the dataset ‘BUPA,’ Lap-MNPSVM preforms obviously worse than other methods. But, from the perspective of average ‘mean’ and ‘std’ accuracy, given at the bottom of Tables 2, 3, 4, our Lap-MNPSVM owns the best performance among all. Lap-MNPSVM has not only the highest average accuracy but also the lowest standard deviation. This shows that Lap-MNPSVM performs better and more stable. In addition to better performance on classification accuracy, our

Table 4 Mean and standard deviation (%) of test accuracy and the number of selected features at 30 % of labeled examples

Dataset	TSVM Acc Std feature	Lap- RLS Acc Std feature	Lap- SVM Acc Std feature	Lap- TSVM Acc Std feature	Lap- MNPSVM Acc Std feature
Hepatitis	78.11	76.75	77.13	79.25	80
	6.73	7.09	8.64	5.74	5.44
	19	19	19	19	18.4
Hearts	71.63	72.39	75.11	76.85	77.53
	6.48	6.71	7.24	6.12	4.04
	14	14	14	14	12.4
Inosphere	69.79	73.24	74.03	75.67	83.46
	7.64	6.59	7.38	5.88	3.43
	34	34	34	34	27
WDBC	91.28	95.89	92.68	94.34	95.76
	6.87	5.14	5.29	5.99	1.21
	30	30	30	30	25.8
Australian	65.73	62.53	66.84	68.48	84.76
	7.64	8.61	5.35	5.63	1.01
	14	14	14	14	13.6
BUPA	63.92	69.48	66.10	68.95	64.47
	7.12	6.29	6.97	7.33	4.69
	6	6	6	6	6
Housevotes	93.12	90.28	92.62	94.01	95.69
	7.11	6.73	7.13	5.34	1.5
	16	16	16	16	16
CMC	62.13	58.21	63.79	64.82	77.55
	7.26	6.43	6.25	5.43	0.1
	9	9	9	9	9
German	61.92	62.92	64.31	63.49	74.13
	8.17	6.47	6.49	8.37	1.66
	20	20	20	20	20
Ave.mean	73.07	73.52	74.73	76.2	81.48
Ave.std	7.22	6.67	6.74	6.20	2.56

Ave.mean and Ave.std denote the average mean and standard deviation accuracy of each algorithm over all datasets

Lap-MNPSVM can pick out the really relevant features, while the other methods can not. The reason may be that the Lap-MNPSVM considers two aspects simultaneously: classification and feature selection. By selecting the proper parameters in the Lap-MNPSVM model, it can balance the classification and feature selection better. In one word, our Lap-MNPSVM performs better than other methods.

3.1.2 Comparison on MNIST dataset

MNIST Dataset is a handwritten digit dataset and consists of gray scale images of handwritten digits from ‘0’ to ‘9’ as shown in Fig. 1. The size of each sample is 28 × 28 pixels.

As the reference [28], we use digits 5 and 8 to form a binary classification problem. The size of labeled examples are 300, 600, and 1,200 separately ('5' and '8' have the same number of samples.) Then, 420 '5' and '8' are randomly selected as unlabeled examples. All labeled and

unlabeled examples together form the training set. The test dataset contains 1,500 examples. We use the training set to select the optimal parameters and learn the classifier and then test it on the test set. The experiments are repeated 10 times. Table 5 shows the result in the case of RBF kernel. We can see that Lap-MNPSVM has not only the best accuracy but also the best stability. The accuracy of Lap-MNPSVM is about 1 % higher than Lap-TSVM in various situations, and the standard deviation is only 0.4 at most. The reason of better performance of our Lap-MNPSVM may be that it has an adaptive norm which can be adjusted according to the dataset while the other methods has only the fixed norm for all datasets.

3.1.3 Comparison on synthetic dataset

In this subsection, we compare the effectiveness of our Lap-MNPSVM with TSVM, Lap-SVM and Lap-TSVM for two semi-supervised synthetic datasets, in terms of the classification performance and decision boundary. Figure 2 shows the one-run results from this dataset of each classifier. It can be found that the decision boundary of TSVM, Lap-SVM and Lap-TSVM is too close to some positive



Fig. 1 An illustration of 10 subjects in MNIST dataset

Table 5 The testing accuracy of '5' versus '8' dataset. The size of the unlabeled data is 420

Labeled data size	TSVM	Lap-SVM	Lap-TSVM	Lap-MNPSVM
300	91.68 ± 2.25	94.37 ± 2.14	95.53 ± 1.98	96.44 ± 0.41
600	92.48 ± 1.88	95.41 ± 2.32	96.34 ± 2.54	97.47 ± 0.3
1,200	93.77 ± 1.45	96.56 ± 1.12	97.23 ± 2.33	98.29 ± 0.17

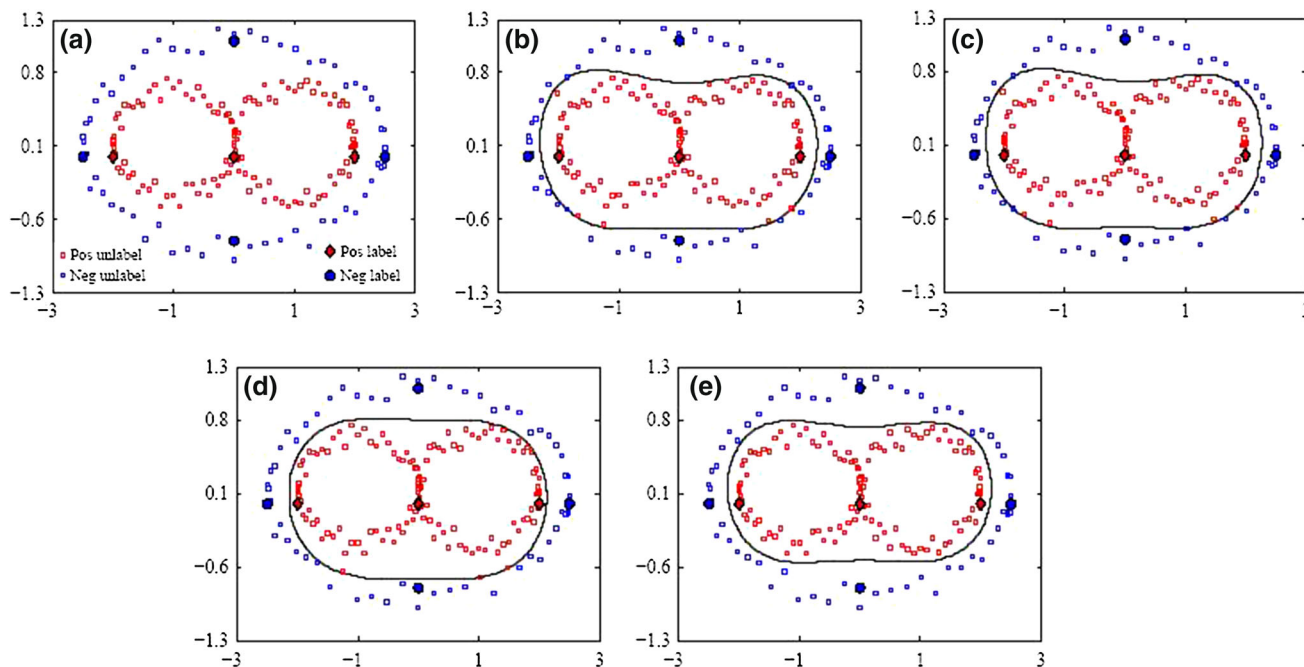


Fig. 2 The comparison on synthetic dataset. a Trueclass, b TSVM, c Lap-SVM, d Lap-TSVM, e Lap-MNPSVM

Table 6 Results on synthetic dataset

Dataset	TSVM	Lap-SVM	Lap-TSVM	Lap-MNPSVM
Three-ellipses	0.9781	0.9726	0.9836	1.0000
	0.1390	0.1103	0.0405	0.0043

examples, while our Lap-MNPSVM can obtain a more reasonable decision boundary than the others. Furthermore, from Table 6, we can see that our Lap-MNPSVM performs best because only it can achieve 100 % accuracy. The behind reason may be that our Lap-MNPSVM has an adaptive norm, which can be adjusted according to the dataset.

4 Conclusion

This paper proposes a novel algorithm named Laplacian mixed-norm proximal support vector machine (Lap-MNPSVM). For the corresponding optimization problem, an approximate local optimal solution is obtained by solving a series of LEs. Furthermore, we analyzing the lower bounds theoretically, which is extremely helpful to select the really relevant features. Our Lap-MNPSVM can automatically realize feature selection and classification for semi-supervised learning. Numerical experiments show that our Lap-MNPSVM is effective in both selecting relevant features and improving the classification accuracy, compared with some popular methods. In the future, one possible work will be extending Lap-MNPSVM to multi-class classification.

Acknowledgments This paper was supported by National Natural Science Foundation of China (No. 11301535, No. 11371365).

References

- Chapelle O, Schölkopf B, Zien A (2010) Semi-supervised learning. MIT Press, MA
- Zhu X, Ghahramani Z, Lafferty Z (2003) Semi-supervised learning using Gaussian fields and harmonic functions. In: Proceedings of the 20th international conference on machine learning (ICML), pp 912–919
- Zhou Z-H, Li M (2010) Semi-supervised learning by disagreement. *Knowl Inf Syst* 24:415–439
- Wang Y, Chen S, Zhou Z (2012) New semi-supervised classification method based on modified cluster assumption. *IEEE Trans Neural Netw Learn Syst* 23:689–702
- Culp M, Michailidis G (2008) Graph-based semisupervised learning. *IEEE Trans Pattern Anal Mach Intell* 30:174–179
- Zhu X, Goldberg AB (2009) Introduction to semi-supervised learning. Morgan and Claypool Press, New York, p 25
- Yang ZX (2013) Nonparallel hyperplanes proximal classifiers based on manifold regularization for labeled and unlabeled examples. *Int J Pattern Recognit Artif Intell* 27(5):1350015
- Ren X, Wang Y, Wang J, Zhang XS (2012) A unified computational model for revealing and predicting subtle subtypes of cancers. *BMC Bioinform*. doi:10.1186/1471-2105-13-70
- Ren XW, wang Y, Zhang XS, Qin J (2013) iPcc: a novel feature extraction method for accurate disease class discovery and prediction. *Nucleic Acids Res* 41(4):e143
- Zheng X, Wu LY, Zhou XB, Stephen TC (2010) Wong Semi-supervised drug-protein interaction prediction from heterogeneous biological spaces. *BMC Syst Biol*. doi:10.1186/1752-0509-4-S2-S6
- Belkin M, Niyogi P (2003) Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput* 15:1373–1396
- 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
- Zhang S, Lei Y, Wu Y (2011) Semi-supervised locally discriminant projection for classification and recognition. *Knowl Based Syst* 24:341–346
- Xue H, Chen S, Yang Q (2009) Discriminatively regularized least-squares classification. *Pattern Recogn* 42:93–104
- Fung G, Mangasarian OL (2001) Proximal support vector machine classifiers. In: Proceedings of international conference of knowledge discovery and data mining, pp 77–86
- Wu F, Wang W, Yang Y, Zhuang Y, Nie F (2010) Classification by semisupervised discriminative regularization. *Neurocomputing* 73:1641–1651
- Chen XJ, Xu FM, Ye YY (2009) Lower bound theory of nonzero entries in solutions of l_2-l_p minimization. <http://www.polyu.edu.hk/ama/staff/xjchen/cxyfinal>
- Bruckstein AM, Donoho DL, Elad M (2009) From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Rev* 51:34–81
- Fan J, Li R (2001) Variable selection via nonconcave penalized likelihood and its oracle properties. *J Am Stat Assoc* 96:1348–1360
- Xu Z, Zhang H, Wang Y, Chang X (2009) $L_{\frac{1}{2}}$ regularizer. *Sci in China Ser F Inf Sci* 52:1–9
- Chen WJ, Tian YJ (2010) L_p -norm proximal support vector machine and its applications. *Procedia Comput Sci ICCS* 1(1):2417–2423
- Tian YJ, Yu J, Chen WJ (2010) L_p -norm support vector machine with CCCP. In: Proceedings of the 7th FSKD, pp 1560–1564
- Tan JY, Zhang CH, Deng NY (2010) Cancer related gene identification via p -norm support vector machine. In: The 4th international conference on computational systems biology, pp 101–108
- Tan J-Y, Zhang Z-Q, Zhen Ling, Zhang C-H, Deng N-Y (2013) Adaptive feature selection via a new version of support vector machine. *Neural Comput Appl* 23:937–945
- Zhang C-H, Shao Y-H, Tan J-Y, Deng N-Y (2013) A mixed-norm linear support vector machine. *Neural Comput Appl* 23:2159–2166
- Saad Y (2003) Iterative methods for sparse linear systems. SIAM Press, Philadelphia
- Joachims T (2002) Learning to classify text using support vector machines: methods, theory and algorithms. Springer, New York
- Qi Z, Tian Y, Shi Y (2012) Laplacian twin support vector machine for semisupervised classification. *Neural Netw* 35:46–53