Journal of Computers Vol. 29 No. 1, 2018, pp. 63-75
doi:10.3966/199115992018012901006

Proximal Support Vector Machine with Mixed Norm

Zhi Li[1], Jun-Yan Tan[2*], Yong-Ning Zhao[1], Lin Ye[1], Rui-Kun Ma[2]

[1] College of Information and Electrical Engineering, China Agricultural University, Beijing, China, 10083
{lizhi_0801, zyn, yelin}@cau.edu.cn

[2] College of Science, China Agricultural University, Beijing, China, 10083
tanjunyan0@126.com, mrk@cau.edu.cn

Received 6 July 2016; Revised 17 February 2017; Accepted 26 March 2017

Abstract. This paper proposes a new version of support vector machine (SVM) for binary classification named *mixed* norm proximal support vector machine, MPSVM for short. By introducing the p-norm of the normal vector of the classification hyper-plane into the objective function of proximal SVM, we get the objective function of MPSVM. MPSVM is an adaptive learning procedure with p-norm ($0 < p < 1$), where p can be automatically chosen by data. By adjusting the parameter p, MPSVM can realize feature selection and classification simultaneously. Since the optimization problem of MBPSVM is neither convex nor differentiable, an iterative algorithm is used to solve it. Experiments carried out on several standard UCI datasets show a clear improvement over some popular methods.

Keywords: binary classification, feature selection, nonlinear classification, p-norm, proximal support vector machine

1 Introduction

It is well known that SVMs have superior performances in classification problem and attracts the researcher's interest [1-4]. Recently, a lot of modifications of SVMs have been made to improve its performance. During those modifications, the lease square SVMs (LS-SVMs) and proximal SVMs received more attention [5-8]. Because LS-SVMs and proximal SVMs only solve a series linear equations while the SVMs have to solve a quadratic programming. And the classification results of LS-SVMs and PSVM are comparable with the standard SVMs [7-11]. Our new method is based on PSVM, the adaptive penalty is introduced into the optimization problem of PSVM. Our new method inherits the advantage of PSVM and can realize feature selection and classification simultaneously.

Feature selection based on support vector machine has been attracted more and more attention [12-18]. Because this kinds of methods have obvious benefits in terms of data storage, computational requirements, and cost of future data collection and they often provide better model understanding. There are several feature selection methods based on SVM. Such as l_0 SVM which replaces the l_2-norm of w by l_0-norm. The optimization problem is hard to be solved due to its incontinuity. In order to overcome the shortcoming of l_0-SVM, Li et al. [12] proposes l_1-SVM which replaces the l_2-norm of w by l_1-norm. The optimization problem of l_1-SVM can be converted to linear programming. Although l_0-SVM and l_1-SVM can realize feature selection and classification simultaneously, they use fixed norm for all kinds of data.

Recently, p-norm ($0 < p < 1$) attracts great attention in the optimization community because using p-norm can have more sparse solutions [7-8, 10]. In the proposed p-norm ($0 < p < 1$) SVMs, the 2-norm penalty in the standard linear SVM is replaced by the p-norm penalty. Compared with the standard SVMs, p-norm SVMs can realize feature selection and classification simultaneously by adjusting the value of p. The l_p-norm is also used in regression problem. [19] proposed a l_p-norm support vector regression (SVR), the l_2-norm penalty is replaced by l_p-norm penalty in the optimization problem of the standard l_2-norm SVR. This paper proposes *mixed*-norm proximal SVM(MPSVM), for the linear situation, the p-norm of

* Corresponding Author

w ($0 < p < 1$) is introduced into the objective function of the primal problem of the linear proximal SVM; for the unlinear situation, the p-norm of dual variable in introduced into the objective function of the dual problem of the unlinear proximal SVM. Our new model is an adaptive learning procedure with p-norm ($0 < p < 1$), where the best p is automatically chosen by data. The same as other p-norm SVMs [20-28], MPSVM can not only realize feature selection but also improve the classification accuracy by adjusting the parameter p. Unfortunately, the optimization problem of MPSVM is neither convex nor differentiable [25-29]. So, it is different to be solved directly. We propose an algorithm to find its approximate solution via solving a series of systems of linear equations (LEs). And the lower bounds for the absolute value of non-zero components in every local optimal solution are established which are extremely helpful to eliminate the zero components in any numerical solution.

Now, we describe our notation. All vectors are column vectors unless transposed to a row vector by a "T". For a n-dimensional vector x, $[x]_i$ ($i = 1, 2, \ldots\ldots, n$) denotes the i-th element of x, $|x|$ denotes a vector in R^n of absolute value of the components of x.$||x||_p$ denotes the value of $x = (|[x]_1|^p, \ldots\ldots, |[x]_n|^p)^{1/p}$ ($1 > p > 0$).$||x||_0$ denotes the number of the non-zero component of x. For two vector $x = ([x]_1, \ldots\ldots, [x]_n)^T \in R^n$ and $y = ([y]_1, \ldots\ldots, [y]_n)^T \in R^n$, $<x \bullet y>$ indicates the inner product of x and y, $x \otimes y$ generates a new vector with the i-th element $[x]_i[y]_i$, ($i = 1, 2, \ldots\ldots, n$). α donates the l-dimensional vector, $[\alpha]_i$ ($i = 1, 2, \ldots, l$) donates the i-th element of α.

This paper is organized as follows. In section 2, the linear PSVM and nonlinear PSVM are introduced firstly, then we carry out linear MPSVM in detail, including solving and analyzing the involved optimization problem. Finally, the nonlinear MPSVM is carried out. In section 3, the numerical experiments on several UCI data sets are conducted to demonstrate the effectiveness of our methods. We conclude this paper in section 4.

2 Methods

In this section, we first introduce proximal support vector machine (PSVM). Then we describe our linear MPSVM and nonlinear MPSVM.

Consider the supervised classification problem with the training sets T,

$$T = \{(x_1, y_1), \ldots\ldots, (x_l, y_l)\} \tag{1}$$

Where $x_j \in R^n$, $j = 1, 2, \ldots\ldots, l$. $y_j \in \{-1, 1\}$, $j = 1, 2, \ldots\ldots, l$. Denote the inputs of all examples as $X = \{x_i\}^l_{i=1} \in R^{l \times n}$ and each row $x_i \in R^n$ is the in the input of the i-th example. $Y = \{y_i\}^l_{i=1} \in R^{l \times n}$ denotes the outputs of labeled examples. Our goal is to construct a classifier which can realize feature selection and get a better generalization performance.

2.1 The Proximal Support Vector Machine (LPSVM)

Instead of the standard support vector machine (SVM) that classifies the examples by assigning them to one of the two disjoint half spaces in input or feature space, PSVM assign examples to the closer one of the two parallel hyperplanes ($(w \bullet x) + b = 1$ and $(w \bullet x) + b = -1$). Its primal problem is:

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}(\|w\|_2^2 + b^2) + \frac{C}{2}\sum_{i=1}^{l}\xi_i^2, \\ s.t. \quad & y_i(\langle w \cdot x_i \rangle + b) = 1 - \xi_i, i = 1, \cdots, l. \end{aligned} \tag{2}$$

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

The dual problem of proximal SVM is:

$$\min_{\alpha} \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j \left(\left(x_i, x_j\right) + 1\right) + \frac{1}{2C}\sum_{i=1}^{l}\alpha^2 - \sum_{i=1}^{l}\alpha_i \tag{3}$$

Where the $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_l]^T$ is the dual variable, (x_i, x_j) is the inner product of x_i and x_j.

Generalize the inner product of xi and x_j to the general kernel $K(x_i, x_j)$, we get the nonlinear case of PSVM:

$$\min_{\alpha} \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j \left(K\left(x_i, x_j\right)+1\right)+\frac{1}{2C}\sum_{i=1}^{l}\alpha^2-\sum_{i=1}^{l}\alpha_i \tag{4}$$

Where the $K(x_i, x_j)$ is the nonlinear kernel function, $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_l]^T$ is the dual variable.

2.2 The Linear *mixed*-norm Proximal Support Vector Machine (LMPSVM)

The proposed LMPSVM is introduced in detail in this section. Adding the *p*-norm of *w* to the primal problem of LPSVM, we get the optimization problem of LMPSVM ($0 < p < 1$):

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}(\|w\|_p^p+\|w\|_2^2+b^2)+\frac{C}{2}\sum_{i=1}^{l}\xi_i^2, \\ s.t. \quad & y_i(\langle w\cdot x_i\rangle+b)=1-\xi_i, i=1,\cdots,l. \end{aligned} \tag{5}$$

Where $C > 0$ is the penalty parameter, p ($0 < p < 1$) is an adjustable parameter.

Now we give the geometric interpretation of problem (5). The first term in the objective function of PSVM is the regularizer, minimizing $||w||_p^p$ can get more sparse solution, minimizing $||w||_2^2$ can realize the maximum margin which makes the final classifier having better generalization performance. The second term is the squared loss function, minimizing it is to minimizing the classification error. The constraints means all positive samples should the closer to the hyper plane $(w \bullet x) + b = 1$ and the negative samples should be closer to the hyper plane $(w \bullet x) + b = -1$.

Problem (5) can be rewritten as the following form by substituting the constraints into the objective function:

$$\min_{w,b} \frac{1}{2}(\|w\|_p^p+\|w\|_2^2+b^2)+\frac{C}{2}\sum_{i=1}^{l}\left[y_i-(\langle w\cdot x_i\rangle+b)\right]^2 \tag{6}$$

Notice that the objective function of (6) is neither convex nor differentiable. So, it is difficult to be solved. To overcome the issue of non differentiable, we approximate $\|w\|_p^p=\sum_{i=1}^{n}|[w]_i|^p$ by $\sum_{i=1}^{n}([w]_i+\varepsilon)^p$, here $\varepsilon > 0$ is a very small number. Thus, problem (6) is approximated by

$$\min_{w,b} F_p(w,b)=\frac{1}{2}\left(\sum_{i=1}^{n}\left(|[w]_i|+\varepsilon\right)^p+\|w\|_2^2+b^2\right)+\frac{C}{2}\sum_{i=1}^{l}[y_i-(\langle w\cdot x_i\rangle+b)]^2 \tag{7}$$

It's clear that the objective function of (7) is differentiable, but still non-convex due to the term $\sum_{i=1}^{n}([w]_i+\varepsilon)^p$ ($0 < p < 1$). To solve this issue, the convex term $\frac{1}{2}\|\beta\otimes w\|_2^2$ is used to approximate the concave term $\sum_{i=1}^{n}(|[w]_i|+\varepsilon)^p$, here β is adjustable to fit the approximation. Then, we get the following convex quadratic program:

$$\min_{w,b} F_2(w,b)=\frac{1}{2}(\|\beta\otimes w\|_2^2+\|w\|_2^2+b^2)+\frac{C}{2}\sum_{i=1}^{l}\left[y_i-(\langle w\cdot x_i\rangle+b)\right]^2 \tag{8}$$

In this paper, β is adjusted iteratively for better approximation. Select an initial $\beta^{(0)} = (\beta_1^{(0)}, \ldots., \beta_n^{(0)})^T$, solve the problem (8) with $\beta^{(0)}$and get solution $(w^{(k)}, b^{(k)})$. In order to get a better approximation, the objective functions F_p and F_2 are required to have the same steepest descent direction at the current $(w^{(k)}, b^{(k)})$, i.e.

$$\nabla F_p(w^{(k)},b^{(k)})=\nabla F_2(w^{(k)},b^{(k)}) \tag{9}$$

$$\Rightarrow p(|[w^{(k)}]_i|+\varepsilon)^{p-1}sign([w^{(k)}]_i)=[\beta_i^{(k+1)}]^2[w^{(k)}]_i \tag{10}$$

$$\Rightarrow \beta_i^{(k+1)} = \sqrt{p(|[w^{(k)}]_i| + \varepsilon)^{p-2}} \quad \textbf{(11)}$$

At the *k*-th iteration, problem (8) is rewritten as the following form:

$$\min_{\tilde{w}} \frac{1}{2}\left[\tilde{w}^T B\tilde{w} + C(\tilde{w}^T \bar{X}_l^T \bar{X}_l \tilde{w} - 2\tilde{w}^T \bar{X}_l^T Y_l + Y_l^T Y_l)\right] \quad \textbf{(12)}$$

Where

$$\bar{X}_l = \begin{pmatrix} x_1^T & 1 \\ x_2^T & 1 \\ \vdots & \vdots \\ x_l^T & 1 \end{pmatrix}_{l\times(n+1)}, \quad \tilde{w} = [w^T \quad b]^T, \quad Y_l = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_l \end{pmatrix}_{l\times 1}, \quad B = diag([\beta^{(k)}]_1^2 + 1, \cdots, [\beta^{(k)}]_n^2 + 1, 1)_{(n+1)\times(n+1)}$$

Note that (12) is an unconstrained quadratic program. Its' KKT conditions lead to solving the following linear equations:

$$(B + C\bar{X}_l^T \bar{X}_l)\tilde{w} = \bar{X}_l^T Y_l \quad \textbf{(13)}$$

The above system of linear equations can be effectively solved by conjugate gradient method (CG).

$A = B + C\bar{X}_l^T \bar{X}_l$ is a symmetric and positive definite matrix, and $b = \bar{X}_l^T Y_l$ is a vector. The equation (13) is then transformed to $A \cdot \tilde{w} = b$ and is solved by the Algorithm 1.

Algorithm 1: the conjugate gradient algorithm for problem (13)

Input: the matrix *A* and vector *b*; the prescribed convergence constant ε_0 ($0 < \varepsilon_0 < 1$); the approximate initial solution $\tilde{w}_0$.

Step 1: Compute the residue vector $r_0 = A\tilde{w}_0 - b$ and the search direction $d_0 = -r_0$, the number of iterations *k* by *k:=0*.

Step 2: If $\|r_k\| \le \varepsilon$, stop the iteration and go to step 5; else go to step 3.

Step 3: Set the step scalar $\alpha_k = \frac{r_k^T r_k}{d_k^T A d_k}$. Update the new solution $\tilde{w}_{k+1} = \tilde{w}_k + \alpha_k d_k$. Renew residual vector $r_{k+1} = r_k + \alpha_k A d_k$ The next step scalar $b_k = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$, the new search direction is $d_{k+1} = -r_{k+1} + b_k d_k$.

Step 4: Increase iterator *k*: = *k* + 1, turn to step 2.

Step 5: Output optimal solution $(w^*, b^*) = \tilde{w}^k$ of the problem (13).

As has been discussed above, we hope we can get the optimal solution of (6). But the feature selection is based on finding the nonzero components of *w**. It's hard to find and identify the real nonzero component of the solution. So we prove the following theorem which is able to identify nonzero components in any local optimal solutions from an approximate local optimal solution and is helpful for feature selection.

Theorem 1 For any local optimal solution (*w**, *b**) to the problem (6), we have $[w^*]_i = 0$. if $[w^*]_i \in (-L_i, L_i)$, where $L_i = \left[\frac{p}{2\left|C\sum_{j=1}^{l}[x_j]_i (b^* - y_j)\right|}\right]^{1/1-p}$, *i* = 1, 2, ..., *n*.

Proof: Suppose $\|w^*\|_0 = k$. Without loss of generality, let $w^* = ([w^*]_1, ..., [w^*]_k, 0, ..., 0)^T$ and $z^* = ([w^*]_1, ..., [w^*]_k)^T$. Construct the new training set $\tilde{T} = \left\{\left(\tilde{x}_1, y_1\right), \cdots, \left(\tilde{x}_l, y_l\right)\right\}$,

Where $\tilde{x}_i = \left([x_i]_1, [x_i]_2, \cdots, [x_i]_k\right)^T \in R^k$, We consider the following optimization problem

$$\min_{z,b} F(z,b) = \frac{1}{2}(\|z\|_p^p + \|z\|_2^2 + b^2) + \frac{C}{2}\sum_{i=1}^{l}[y_i - (\langle z \cdot \tilde{x}_i \rangle + b)]^2$$

$$= \frac{1}{2}(\|z\|_p^p + \|z\|_2^2 + b^2) + \frac{C}{2}(z^T \tilde{X}_l^T \tilde{X}_l z + 2z^T \tilde{X}_l^T (eb - Y) - 2Y^T eb + b^2 e^T e + Y^T Y) \quad (14)$$

$$= \frac{1}{2}(\|z\|_p^p + \|z\|_2^2 + b^2) + \frac{C}{2}(\sum_{i=1}^{k}\lambda_i z_i^2 + 2z^T \tilde{X}_l^T (eb - Y) - 2Y^T eb + b^2 e^T e + Y^T Y)$$

Where

$$\tilde{X}_l = \begin{pmatrix} \tilde{x}_1^T \\ \tilde{x}_2^T \\ \vdots \\ \tilde{x}_l^T \end{pmatrix}_{l\times k}, \qquad e = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}_{l\times 1}, \qquad Y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_l \end{pmatrix}_{l\times 1},$$

$\lambda_1 \geq 0, \ldots, \lambda_k \geq 0$ are the eigenvalues of semi-definite matrix $\tilde{X}_l^T \tilde{X}_l$ and it is easy to know that (z^*, b^*) is a local optimal solution of (14), according to the KKT conditions, we have

$$\nabla F(z^*, b^*) = 0 \quad (15)$$

That is to say,

$$\left(\frac{1}{2}p\left|\left[z^*\right]_i\right|^{p-1} + \left|[z^*]_i\right| + C\lambda_i\left|\left[z^*\right]_i\right|\right)\bullet \operatorname{sgn}\left(\left[z^*\right]_i\right) = \left[-C\tilde{X}_l^T(eb^* - Y)\right]_i$$

$$= -C\sum_{j=1}^{l}\left[\tilde{x}_j\right]_i (b^* - y_j), i = 1, \cdots, k. \quad (16)$$

Take the absolute value on both sides of (16), we have

$$\frac{1}{2}p\left|\left[z^*\right]_i\right|^{p-1} + \left|[z^*]_i\right| + C\lambda_i\left|\left[z^*\right]_i\right| = \left|C\sum_{j=1}^{l}\left[\tilde{x}_j\right]_i (b^* - y_j)\right|, i = 1, \cdots, k. \quad (17)$$

Since $\lambda_i \geq 0$, $i = 1, \ldots\ldots, k.$ it is easy to have

$$\frac{1}{2}p\left|\left[z^*\right]_i\right|^{p-1} \leq \frac{1}{2}p\left|\left[z^*\right]_i\right|^{p-1} + \left|[z^*]_i\right| + C\lambda_i\left|\left[z^*\right]_i\right| = \left|C\sum_{j=1}^{l}\left[\tilde{x}_j\right]_i (b^* - y_j)\right| \quad (18)$$

So,

$$\frac{1}{2}p\left|\left[z^*\right]_i\right|^{p-1} \leq \left|C\sum_{j=1}^{l}\left[\tilde{x}_j\right]_i (b^* - y_j)\right|, \quad (19)$$

Which is equivalent to $\left|\left[z^*\right]_i\right| \geq \left[\frac{p}{2\left|C\sum_{j=1}^{l}\left[\tilde{x}_j\right]_i (b^* - y_j)\right|}\right]^{1/1-p}$ $i = 1, 2, \ldots\ldots, k.$ Note that, $\left[\tilde{x}_j\right]_i = \left[x_j\right]_i$. Define

$L_i = \left[\frac{p}{2\left|C\sum_{j=1}^{l}\left[x_j\right]_i (b^* - y_j)\right|}\right]^{\frac{1}{1-p}}$.

The inequality (19) is indeed $|[z^*]| \geq L_i$, $i = 1, \ldots\ldots, k$. This means that for any nonzero component $[w^*]_i$ of w^*, it satisfies$|[w^*]_i| \geq L_i$, $i = 1, 2, \ldots\ldots, n$. Equivalently, for any local optimal solution (w^*, b^*) of (8), we have$[w^*]_i \in (-L_i, L_i) \Rightarrow [w^*]_i = 0$, $i = 1, 2, \ldots\ldots, n.$

Based on the discussion, the following algorithm is established.

Algorithm 2: linear mixed-norm proximal support vector machine (LMPSVM)

Input: the training set (1), parameters $C(C > 0)$, $p(0 < p < 1)$; a specified maximum number of iterations K and a very small $\varepsilon_1 \geq 0$.

Step 1: Given a random vector β_0, generate the optimization problem (6).

Step 2: Use Algorithm 1 to solve problem (13), get the solution (w^*, b^*) and update $\beta^{(k+1)}$ by the expression (11).

Step 3: Terminate the operations and proceed to next step until $\|\beta^{(k+1)}-\beta^{(k)}\| < \varepsilon_1$ or iterations $k > K$. Or let $k = k+1$and go to step 2.

Step 4: Output the optimal solution (w^*, b^*) of the problem (6). Output classification hyperplane $(w^*\cdot x)+b^* = 0$ and feature subset $F^{'} = \{i|[w^*]_i > L_i, i = 1, 2 \ldots, n\}$;

2.3 The Nonlinear *mixed*-norm Proximal Support Vector Machine (NMPSVM)

By adding the p-norm of α to the dual problem of nonlinear PSVM, we get the optimization problem of NMPSVM:

$$\min_{\alpha} \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j \left(K\left(x_i, x_j\right)+1\right)+\frac{1}{2C}\sum_{i=1}^{l}\alpha^p+\frac{1}{2C}\sum_{i=1}^{l}\alpha^2-\sum_{i=1}^{l}\alpha_i \tag{20}$$

Where the $K(x_i, x_j)$ is the nonlinear kernel function. $C > 0$ is the parameter.

Similar as the LMPSVM, the optimization problem of NMPSVM is neither convex nor differentiable because of the item $\sum_{i=1}^{l}\alpha^p$. The $\sum_{i=1}^{l}\alpha^p$ is approximated by $\sum_{i=1}^{l}\left(\alpha+\varepsilon\right)^p$ to make itself differentiable. And the $\sum_{i=1}^{l}\left(\alpha+\varepsilon\right)^p$ is approximated by $\|\beta\otimes\alpha\|_2^2$ to make it convex. The formula (20) can be written approximately as follows:

$$\min_{\alpha} \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j \left(K\left(x_i, x_j\right)+1\right)+\frac{1}{2C}\|\beta\otimes\alpha\|_2^2+\frac{1}{2C}\sum_{i=1}^{l}\alpha^2-\sum_{i=1}^{l}\alpha_i \tag{21}$$

The matrix form of (21) can be shown as below:

$$\min_{\alpha} \alpha^T LK\left(X_l, X_l^T\right)L^T\alpha+\alpha^T LL^T\alpha+\frac{1}{C}\alpha^T B\alpha+\frac{1}{C}\alpha^T\alpha-\alpha^T I \tag{22}$$

Where

$$\alpha=\begin{pmatrix}\alpha_1\\ \alpha_2\\ \vdots\\ \alpha_l\end{pmatrix}_{l\times 1}, L = diag\left(Y_l\right), Y_l=\begin{pmatrix}y_1\\ y_2\\ \vdots\\ y_l\end{pmatrix}_{l\times 1}, X_l=\begin{pmatrix}x_1^T\\ x_2^T\\ \vdots\\ x_l^T\end{pmatrix}_{l\times n}, I=\begin{pmatrix}1\\ 1\\ \vdots\\ 1\end{pmatrix}_{l\times 1}, B = diag([\beta^{(k)}]_1^2+1,\cdots,[\beta^{(k)}]_n^2+1,1)_{(l+1)\times(l+1)}$$

By the KKT of the equation (22), we get the following equation:

$$\left(LK\left(X_l, X_l^T\right)L^T+LL^T+\frac{1}{C}B+\frac{1}{C}\right)\alpha=I \tag{23}$$

The nonlinear classification is solved by the similar method to that used for linear classification. In the nonlinear KKT (23), $A = LK\left(X_l, X_l^T\right)L^T+LL^T+\frac{1}{C}B+\frac{1}{C}$, $b = I$. The equation (23) is then transformed to $A\cdot\alpha = b$ and is solved by the Algorithm 1 and Algorithm 2.

Finally the separation hyperplane obtained from nonlinear model is:

$$\sum_{i=1}^{l}\alpha_i^* y_i K(x_i, x)+\sum_{i=1}^{l}y_i\alpha_i^* = 0 \tag{24}$$

Where the α^* is the solution of nonlinear model solved by Algorithm 1 and Algorithm 2.

3 Experiment and Result Analysis

The accuracy Acc[1] of the classifier is a measure of the merits of the algorithm. We compared MPSVM with others classifiers, including 1-norm SVM, *p*-norm SVM, and PSVM on several UCI data sets. The following is a brief introduction of them:

- 1-norm SVM [29]: the 2-norm of normal vector in the standard SVM is replaced by its 1-norm. This simple modification can realize classification and feature selection simultaneously.
- p-norm SVM [21]: the 2-norm of normal vector in the standard SVM is replaced by its p-norm ($p \in [0,1]$. p-norm SVM performs better than 1-norm SVM because p-norm SVM has both more sparsity and better accuracy.
- PSVM: The optimization problem of PSVM is an unconstrained quadratic program and can be easily solved by solving a series of linear equations.

The experiments are conducted on UCI datasets and synthetic datasets. On UCI datasets, we compare the classification performance of MPSVM and the other three classifiers we mentioned above. On synthetic dataset, we analyze the impact of p on the feature selection performance of MPSVM.

Our experiments are carried out on the platform of MATLAB 2010 on PC with an Intel Core I3 processor and 2GB RAM. With regard to parameter selection, the 10-fold cross-validation technique is used in the training procedure. Parameter C and the kernel parameter sigma is all selected from the set $\{2^i | i = -5, -4... 5\}$. The parameter p of MPSVM is selected from the set {0.1, 0.2, …, 0.9}.

3.1 Comparison on UCI Datasets

We use 10 UCI data sets which are frequently used in binary classification problem to compare four classifiers. The statistical characters of the data sets rare listed in Table 1. We can see that the number of features varies from 6 to 44 and the samples size ranges from 155 to 1473.

Table 1. Summary of UCI datasets

Datasets	Size	features
Australian	690	14
BUPA	345	6
CMC	1473	9
German	1000	20
Hearstatlog	270	13
hepatitis	155	19
Hourse	300	26
Ionosphere	351	33
Spect	267	44
wdbc	569	30

Comparison LMPSVM vs. the other three classifiers. To avoid the disturbance brought by different magnitudes of datasets, all datasets are normalized into range of [-1, 1] before training. The 10-fold cross-validation accuracies of four methods are compared. Each dataset is randomly divided into ten parts, nine of them are used as the training set and the remaining one is used as the test set. The training set is used to build the classifiers and the test set is used to test the classification performance. We perform 10 times ten-fold cross validation and the average accuracy is used. The accuracy is more closed to 100, the classifier is more better. Table 2 shows experimental results, including the average accuracy, standard deviation of accuracy, the number of feature selected by different classifiers. The best result is marked by bold-face.

[1] Accuracy (Acc) is utilized to evaluate the performance of classification and is defined as follow. Accuracy denotes the percentage of both positive points and negatives points correctly predicted and is defined as follows:
Acc=TP+TN/TP+TN+FP+FN,
Where TP, TN, FP and FN denotes the number of true positives, true negatives, false positives and false negatives, respectively.

From Table 2, it can be seen that the classification performance of MPSVM is better than other methods on all datasets. The average accuracy of MPSVM is 85.573% which is 2.4% higher than the other three methods. The average standard deviation of MPSVM is smallest, revealing the stable performance of MPSVM in classification. Meanwhile, the feature selection is accomplished. We can see that MPSVM select less features than the other three methods. In one words, MPSVM performs better on both classification and feature selection.

Table 2. Mean (%) and standard deviation (‰) of test accuracy. Ave.mean and Ave.std denotes the average mean and standard deviation accuracy of each algorithm over all datasets

Dataset	1-norm SVM Acc Std feature	P-norm SVM Acc Std feature	PSVM Acc Std feature	MPSVM Acc Std feature
Australian	86.44 6.1 14	86.11 6.6 14	86.95 7.7 14	**88.96** 0.22 **12.6**
BUPA	70.01 4.5 6	65.31 13.9 6	70.47 20.3 6	**74.52** 4.51 6
CMC	77.74 0.28 8.5	77.46 0.55 **7.2**	77.66 1.5 9	**77.99** 0 **7.7**
German	76.27 3.7 20	75.27 3.7 20	76.14 6.0 20	**78.26** 0.35 **18.6**
Heartstatlog	84.85 6.2 13	85.53 5.0 13	85.52 15.5 13	**88.56** 0.11 **12.9**
hepatitis	85.37 6.0 19	84.76 5.8 18.9	86.39 8.2 19	**90.41** 0.45 **17.5**
Ionosphere	89.95 25.9 32.8	88.93 28.9 33	87.94 10.3 33	**90.83** 0.17 **30.2**
Hourse	82.64 8.7 25.9	83.41 25.1 24.2	83.68 35.8 26	**86.63** 0.2 **21.8**
Spect	80.78 7.8 44	80.56 5.3 43.9	79.74 7.4 44	**81.20** 0.17 **35.1**
wdbc	97.41 3.0 29.7	98.10 1.2 28.8	96.49 3.7 30	**98.37** 0.07 **21.8**
Ave.mean	83.146	82.543	83.098	**85.573**
Ave.std	9.888	9.605	10.87	0.625

Comparison of LMPSVM vs. the RFE-LPSVM approach. Although the LMPSVM can realize feature selection and classification simultaneously and it performs better than other classifiers, how it compares with the methods that combine feature selection and classification? Thus, to evaluate the performance of MPSVM more reasonably, we compare LMPSVM with the methods that a preceding feature selection conducted and the selected features are used for the classifier.

Because LMPSVM is based on PSVM, we compare LMPSVM with 2s-LPSVM. 2s-LPSVM is conducted as follows: firstly, a feature selection method is used to determine the features; secondly, the classification is carried out using the LPSVM on the selected features. The Support Vector Machine Recursive Feature Elimination (SVM-RFE) [3] is used for feature selection in 2s-LPSVM. So we change the name of 2s-LPSVM into RFE-LPSVM. The test results are shown in Table 3. It can be seen that, though the 2s-LPSVM has less selected features than the LMPSVM, but its performance is still worse

than that of MPSVM in terms of both accuracy and stability.

Table 3. Mean (%)and standard deviation (‰) of test accuracy. Ave.mean and Ave.std denotes the average mean and standard deviation accuracy of RFE - PSVM and MPSVM over all datasets

dataset	RFE - LPSVM ACC std feature	LMPSVM ACC std feature
Australian	86.35 3.5 2	**88.96** 0.22 **12.6**
BUPA	68.65 7.1 5	**74.52** 4.51 6
CMC	77.44 0.57 2	**77.99** 0 **7.7**
German	76.67 9.2 16	**78.26** 0.35 **18.6**
Heartstatlog	85.81 5.3 9	**88.56** 0.11 **12.9**
hepatitis	86.12 8.3 7	**90.41** 0.45 **17.5**
Ionosphere	88.34 6.5 17	**90.83** 0.17 **30.2**
Hourse	82.97 11.3 2	**86.63** 0.2 **21.8**
Spect	79.78 2.8 2	**81.20** 0.17 **35.1**
wdbc	96.03 1.7 8	**98.37** 0.07 **21.8**
Ave.mean	82.816	**85.573**
Ave.std	4.797	0.625

Comparison of computational time. The computational time of classification time is used as a measure of the efficiency of the classifiers. For each dataset, 90% of the dataset is used for model training and 10% of that is used for testing. The four classifiers, including 1-norm SVM, p-norm SVM, RFE-PSVM and LMPSVM, are trained with the same training datasets and tested with the same testing datasets. Finally, the computational time of the four classifiers for each dataset are given logarithmically. Additionally, the average computational time for the 10 datasets for each classifier is shown in Fig. 1.

It can be concluded that the computational time cost by LMPSVM is much less than other classifiers, indicating a higher efficiency of the LMPSVM.

The performance comparison of NMPSVM and NPSVM. In this section, we compare the NMPSVM and nonlinear PSVM on UCI data sets. The Gausse kernel is used in two algorithms. The comparison results are shown in Table 4.

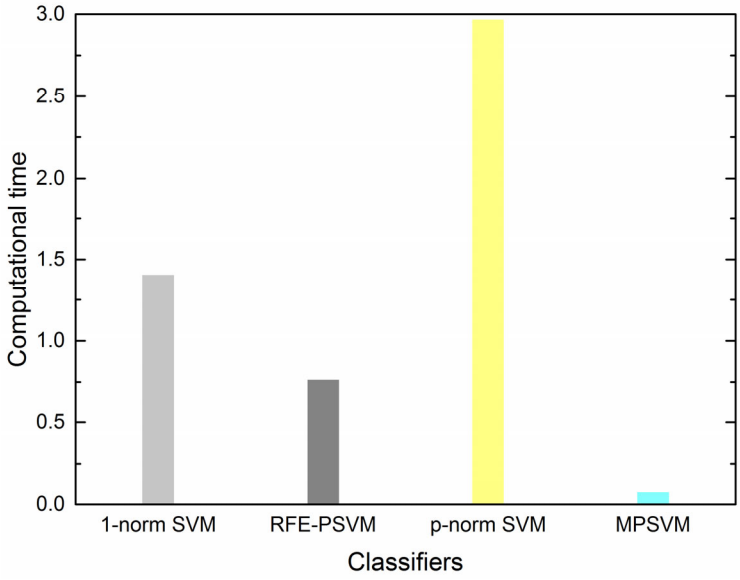

Fig. 1. The average computational time for different classifiers

Table 4. Mean (%) and standard deviation (‰) of test accuracy. Ave.mean and Ave.std denotes the average mean and standard deviation accuracy of the nonlinear of PSVM and MPSVM over all datasets

dataset	NMPSVM ACC std	NPSVM ACC std
Australian	**87.65** 0.30	86.90 0.53
BUPA	**60.12** 0.66	59.99 0.74
CMC	**77.45** 0.07	77.39 0.30
German	**72.22** 0.19	72.00 0.78
Heartstatlog	**85.70** 0.94	85.67 1.02
hepatitis	79.76 0.24	**80.98** 0.38
Ionosphere	**60.48** 0.10	60.46 0.13
Hourse	**73.13** 8.7	36.12 0.22
Spect	**79.99** 0.15	79.91 0.60
wdbc	**63.33** 3.63	37.37 0.07
Ave.mean	**73.983**	67.679
Ave.std	1.498	0.477

Table 4 shows that the classification accuracy of NMPSVM is higher than NPSVM in most cases. Especially on the datasets “Hourse” and “wdbc”, the accuracy is almost twice of NPSVM. The standard deviation of accuracy of algorithm is no more than 10. It means the classification of NMPSVM shows lower volatility. It can be seen the NMPSVM is more robust in terms of std. in general. Through the above analysis, the NMPSVM has better overall performance.

3.2 Comparison on Synthetic Datasets

To analyze the impact of p on feature selection of MPSVM, the MPSVM is tested using a synthetic dataset and five UCI datasets. The synthetic dataset used here is the same with that in the reference [10]. It is randomly generated by the following steps:

(1) The inputs $x_i \in R^n$ are stochastic vectors independently generated from the normal distribution $N(0,1)$, $i = 1, 2, ..., 100,$ and n is set equal to 40.

(2) The outputs are determined by the hyper plane $g(x) = 4[x]_1+2[x]_2+4[x]_3-0.1$, which means that the output of an input x_i is '+1' if $g(x_i) \geq 0$ and is '-1' if $g(x_i) < 0$.

The impact of p on feature selection of LMPSVM. The synthetic datasets are randomly generated; each synthetic dataset and five UCI data sets are randomly divided into two subsets: 90% for training, 10% for testing. For each experiment, we fix the parameter $C = 10^{-5}$ and iteration $K = 100$, using the training set to learn the classifier and test it on the testing dataset. Each experiment is repeated 10 times. We selected the number of selected features in each experiment when p varies from 0.1 to 0.9. Fig. 2 shows the impact of p on different datasets. It can be seen that MPSVM controls sparsity and the number of selected features by adjusting the value of p.

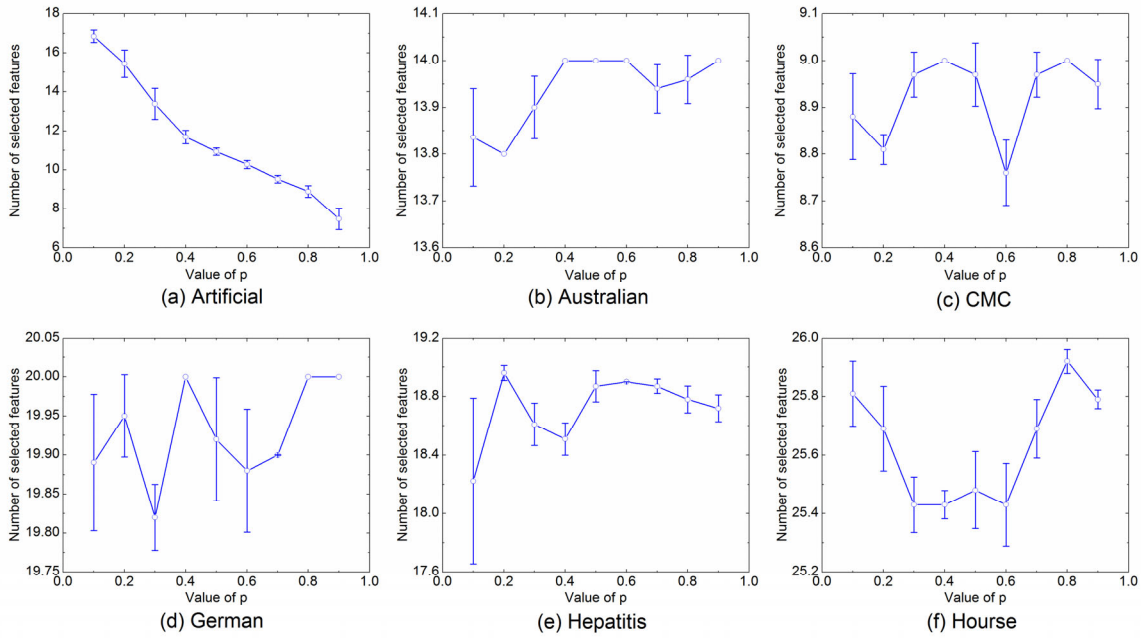

Fig. 2. The impact of p on different datasets

4 Conclusion

This paper proposes two novel algorithms: linear *mixed*-norm proximal support vector machine (LMPSVM) and nonlinear *mixed*-norm proximal SVM (NMPSVM), which can realize the feature selection and classification simultaneously. We can adjust the norm and other parameters to balance feature selection and classification accuracy. The MPSVM can transform the original optimization problem into a differentiable and convex optimization problem, for which an approximate local optimal solution can be obtained by solving a series of linear equations. Experiments demonstrate that our approach performs better in terms of both classification accuracy and efficiency in comparison with other classifiers. Furthermore, the lower bounds of absolute of the approximate solution are constructed which are helpful for feature selection. All in all, the proposed method presents better classification and feature selection in the binary classification.

Acknowledgements

This paper is supported by the National Nature Science Foundation of China (Grant No.11301535, 11371365) and the Chinese Universities Scientific Fund (Grant No.2017LX003).

References

[1] Y.H. Shao, N.Y. Deng, Z.M. Yang, Least squares recursive projection twin support vector machine for classification, Pattern Recognition 45(6)(2012) 2299-2307.

[2] P. Rebentrost, M. Mohseni, S. Lloyd, Quantum support vector machine for big data classification, Physical Review Letters 113(13)(2014) 130503.

[3] I. Guyon, J. Weston, S. Barnhill, V. Vapnik, Gene selection for cancer classification using support vector machine, Machine Learning 46(1-3)(2002) 389-422.

[4] J. Luts, F. Ojeda, R. Van de Plas, B. De Moor, S. Van Huffel, J.A.K. Suykens, A tutorial on support vector machine-based methods for classification problems in chemometrics, Analytica Chimica Acta 665(2)(2010) 129-145.

[5] Z. Zhang, L. Zhen, N. Deng, J. Tan, Sparse least square twin support vector machine with adaptive norm, Applied intelligence 41(4)(2014) 1097-1107.

[6] Y.-H. Shao, N.-Y. Deng, C.-N. Li, X.-Y. Hua, Robust Lp-norm least squares support vector regression with feature selection, Applied Mathematics and Computation 305(2017) 32-52.

[7] W.J. Chen, Y.J. Tian, Lp-norm proximal support vector machine and its applications, Procedia Computer Science 1(1)(2010) 2417-2423.

[8] Y. Tian, J. Yu, W. Chen, l p-norm support vector machine with CCCP, in: Proc. Fuzzy Systems and Knowledge Discovery (FSKD), 2010 Seventh International Conference on. IEEE 4, 2010.

[9] J.Y. Tan, C.H. Zhang, N.Y. Deng, Cancer related gene identification via p-norm support vector machine, in: Proc. the 4th International Conference on Computational Systems Biology, 2010.

[10] X. Chen, F. Xu, Y. Ye, Lower bound theory of nonzero entries in solutions of \ell_2-\ell_p minimization, SIAM Journal on Scientific Computing 32(5)(2010) 2832-2852.

[11] A.M. Bruckstein, D.L. Donoho, M. Elad, From sparse solutions of systems of equations to sparse modeling of signals and images. SIAM review 51(1)(2009) 34-81.

[12] C.N. Li, Y.H. Shao, N.Y. Deng, Robust L1-norm non-parallel proximal support vector machine, Optimization 65(1)(2016) 169-183.

[13] E.J. Hess, J.P. Brooks, The support vector machine and mixed integer linear programming: ramp loss SVM with L1-norm regularization, in: Proc. the 14th INFORMS Computing Society Conference, 2015.

[14] H.L. Chen, B. Yang, J. Liu, D.Y. Liu, A support vector machine classifier with rough set-based feature selection for breast cancer diagnosis, Expert Systems with Applications 38(7)(2011) 9014-9022.

[15] S. Li, H. Wu, D. Wan, J. Zhu, An effective feature selection method for hyperspectral image classification based on genetic algorithm and support vector machine, Knowledge-Based Systems 24(1)(2011) 40-48.

[16] S. Maldonado, R. Weber, J. Basak, Simultaneous feature selection and classification using kernel-penalized support vector machines, Information Sciences 181(1)(2011) 115-128.

[17] L. Bai, Z. Wang, Y.H. Shao, N.Y. Deng, A novel feature selection method for twin support vector machine, Knowledge-

Based Systems 59(2014) 1-8.

[18] L. Yao, F. Zeng, D.-H. Li, Z.-G. Chen, Sparse support vector machine with Lp penalty for feature selection, Journal of Computer Science and Technology 32(1)(2017) 68-77.

[19] C. Zhang, D. Li, J. Tan, The support vector regression with adaptive norms, Procedia Computer Science 18(2013) 1730-1736.

[20] C. Zhang, Y. Shao, J. Tan, N. Deng, Mixed-norm linear support vector machine, Neural Computing and Applications 23(7-8)(2013) 2159-2166.

[21] J. Tan, Z. Zhang, L. Zhen, C. Zhang, N. Deng, Adaptive feature selection via a new version of support vector machine, Neural Computing and Applications 23(3-4)(2013) 937-945.

[22] Z. Zhang, T. Ke, N. Deng, J. Tan, Biased p-norm support vector machine for PU learning, Neurocomputing 136(2014) 256-261.

[23] X. Peng, TPMSVM: a novel twin parametric-margin support vector machine for pattern recognition, Pattern Recognition 44(10)(2011) 2678-2692.

[24] Y.H. Shao, N.Y. Deng, A novel margin-based twin support vector machine with unity norm hyperplanes, Neural Computing and Applications 22(7-8)(2013) 1627-1635.

[25] Z.Q. Zhang, T. Ke, N.Y. Deng, J.Y. Tan, Biased p-norm support vector machine for PU learning, Neurocomputing 136(2014) 256-261.

[26] J.Y. Tan, L. Zhen, N.Y. Deng, Z.Q. Zhang, Laplacian p-norm proximal support vector machine for semi-supervised classification, Neurocomputing 144(2014) 151-158.

[27] C.N. Li, Y.H. Shao, N.Y. Deng, Robust L1-norm non-parallel proximal support vector machine, Optimization 65(1)(2016) 169-183.

[28] X. Peng, D. Xu, L. Kong, D. Chen, L1-norm loss based twin support vector machine for data recognition, Information Sciences 340(C)(2016) 86-103.

[29] J. Zhu, S. Rosset, T. Hastie, R. Tibshirani, 1-norm support vector machines, Advances in Neural Information Processing Systems 16(1)(2004) 49-56.