Applied Intelligence (2019) 49:2301–2314
https://doi.org/10.1007/s10489-018-1377-x

All-in-one multicategory Ramp loss maximum margin of twin spheres support vector machine

Sijie Lu[1] · **Huiru Wang**[1] · **Zhijian Zhou**[1]

Published online: 12 January 2019

Abstract
Maximum margin of twin spheres support vector machine (MMTSSVM) is effective to deal with imbalanced data classification problems. However, it is sensitive to outliers because of the use of the Hinge loss function. To enhance the stability of MMTSSVM, we propose a Ramp loss maximum margin of twin spheres support vector machine (Ramp-MMTSSVM) in this paper. In terms of the Ramp loss function, the outliers can be given fixed loss values, which reduces the negative effect of outliers on constructing models. Since Ramp-MMTSSVM is a non-differentiable non-convex optimization problem, we adopt Concave-Convex Procedure (CCCP) approach to solve it. We also analyze the properties of parameters and verify them by one artificial experiment. Besides, we use Rest-vs.-One(RVO) strategy to extend Ramp-MMTSSVM to multi-class classification problems. The experimental results on twenty benchmark datasets indicate that no matter in binary or multi-class classification cases, our approaches both can obtain better experimental performance than the compared algorithms.

Keywords MMTSSVM · Ramp loss · CCCP · Multi-class

1 Introduction

An increasing number of machine learning approaches are prevailing these days, such as artificial neural networks [1], multi-objective optimization [2] and support vector machine (SVM) [3], and they all have their pros and cons. The SVM, proposed by Vapnik, is based on statistical learning theory. Under the guidance of VC dimension and structural risk minimization principle, it can get a good generalization and promotion ability through the compromise between empirical risk and model complexity. It is very effective when solving small sample, nonlinear, high dimensional sample problems, also feasible to avoid dimensional disasters and over-fitting problems to some extent. Nowadays, SVM has been applied to many fields, containing text classification [4], disease detection [5, 6], driver fatigue detection [7] and object detection [8] etc. Nevertheless, there are still some defects in SVM, and many improvements have been put forward in recent years.

One drawback of classic SVM is the slow computational speed. In order to improve this problem, Jayadeva et al. [9] proposed a twin SVM (TSVM) for binary classification problems. TSVM generates two nonparallel hyperplanes by solving two smaller-sized quadratic programming problems (QPPs) rather than one large QPP, which makes the computational speed approximately four times faster than that of classic SVM in theory. Meanwhile, TSVM requires that the data points of one class are as close as possible to one hyperplane and as far as possible from the other. Based on TSVM, many algorithms in [10–17] have been proposed.

In many TSVM-based algorithms, they all need matrix inverse operation. However, sometimes the matrix may not be reversible. What's more, solving a matrix inverse is computationally expensive. In order to remedy this problem, the twin hypersphere support vector machine (THSVM) [18] was raised. It finds two irrelevant hyperspheres rather than two nonparallel hyperplanes by solving a pair of smaller-sized QPPs. It requires each hypersphere to capture as many data points of one class as possible. Because it needs to find two centers and two radiuses, it is still computational cost. In addition, THSVM can not deal with imbalanced data classification problems well.

To decrease computational cost and deal with imbalanced data classification problems more efficently, Xu [19] came up with a maximum margin of twin spheres support

✉ Zhijian Zhou
zzjmath@163.com

[1] College of Science, China Agricultural University, No.17 Qinghua East Road, 100083, Haidian, Beijing, China

vector machine (MMTSSVM). It finds two homocentric hyperspheres rather than two irrelevant hyperspheres by solving one QPP and one linear programming problem (LPP).

All algorithms mentioned above use the Hinge loss function, which makes the models sensitive to outliers. The outliers may normally be given the largest hinge loss values, so the decision hyperplane is drawn toward outliers incorrectly, leading to decreased generalization performance. Hence, Huang et al. [20] proposed a ramp loss support vector machine (RSVM). According to the Ramp loss function, the outliers can be given fixed loss values. Thus, RSVM decreases the sensitivity to outliers. Because the RSVM is a non-convex optimization problem, the authors adopt the effective Concave-Convex Procedure (CCCP) [21] to solve it. Based on RSVM, some algorithms have been proposed, such as RLSSVM [22] etc.

In our daily life, multi-class classification problems are more common. The researchers have proposed a great deal of multi-class classification strategies, such as One-vs.-One (OVO) [23, 24], One-vs.-Rest (OVR) [25, 26], Rest-vs.-One (RVO) [27, 28] and One-vs.-One-vs.-Rest(OVOVR) [29–31]. For a K-class classification problem, OVO needs to combine the K classes in pairs. We choose the ith class as the positive class, and the jth class as the negative class to generate a classifier. Thus, $K(K\text{-}1)/2$ classifiers will finally be obtained. In OVR method, we choose the ith class as the positive class, and the rest classes as the negative class to generate a classifier. In total, OVR will generate K classifiers. RVO is just opposite to OVR. It picks the ith class as the negative class and the rest classes as the positive class to generate a classifier. RVO will also generate K classifiers. OVOVR is a bit similar with OVO. It also combines the classes in pairs. We select the ith class as the positive class, the jth class as the negative class and the remaining classes as the rest class to generate one classifier. OVOVR will also generate $K(K\text{-}1)/2$ classifiers totally. All the strategies above determine the type of a new data point by 'voting' scheme.

In order to improve the generalization performance of MMTSSVM, we propose a Ramp loss maximum margin of twin spheres support vector machine (Ramp-MMTSSVM) in this article. We employ MMTSSVM as the core of our algorithm because it has faster computational speed and can deal with imbalanced data classification problems effectively. Meanwhile, we use the Ramp loss function to substitute the Hinge loss function to decrease the sensitivity of MMTSSVM to outliers. After introducing the Ramp loss function, the optimization problem becomes a non-differentiable non-convex problem, which can not be solved by quadratic programming directly. Therefore, we use CCCP approach to deal with it. We also discuss the properties of parameters in Ramp-MMTSSVM, which can help us determine the range of parameters. In addition, we extend Ramp-MMTSSVM to multi-class classification problems by RVO strategy, termed as multicategory Ramp loss maximum margin of twin spheres support vector machine (MRMMTSSVM). Through twenty benchmark experiments, we compare Ramp-MMTSSVM with THSVM, SSLM and MMTSSVM and compare MRMMTSSVM with OVO-THSVM, THKSVM and OVR-MMTSSVM. The experimental results imply that our algorithms have better performance than other algorithms.

The paper is organized as follows. Section 2 gives a brief introduction on MMTSSVM and RSVM. In Section 3, we introduce Ramp-MMTSSVM. Section 4 shows our algorithm, MRMMTSSVM. Several artificial experiments and twenty benchmark experiments to testify the effectiveness of our algorithms are shown in Section 5. The last section, Section 6, is our conclusion on present study.

2 Background

In this section, we review the basics of MMTSSVM and RSVM.

To make it easier to understand, we first give some definitions of notions. Given a set $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)\}$, where $\mathbf{x}_i \in \mathcal{R}^m$ and $y_i \in \{+1, -1\}$, $i = 1, \ldots, l$ includes l^+ positive data points and l^- negative data points. In addition, I^+ and I^- represent positive set and negative set, respectively. ϕ is a nonlinear mapping, which maps the input data points into the higher-dimensional feature space. Kernel function $K(\mathbf{x}_i, \mathbf{x}_j) = \big(\phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)\big)$ is selected in advance.

2.1 Maximum margin of twin spheres support vector machine

MMTSSVM is good at dealing with imbalanced data classification problems. It aims to find two homocentric spheres. On one hand, the small sphere needs to cover as many data points of the positive class as possible. On the other hand, the large sphere needs to push out as many data points of the negative class as possible. In addition, it follows the maximum margin principle which requires the margin between the small sphere and the large sphere is as large as possible. The optimization problems that MMTSSVM needs to solve are denoted as follows,

$$\begin{aligned} \min_{R^2, C, \xi_i} \quad & R^2 - \frac{\nu}{l^-} \sum_{j \in I^-} \|\phi(\mathbf{x}_j) - C\|^2 + \frac{1}{\nu_1 l^+} \sum_{i \in I^+} \xi_i \\ s.t. \quad & \|\phi(\mathbf{x}_i) - C\|^2 \le R^2 + \xi_i, \\ & \xi_i \ge 0, i \in I^+, \end{aligned} \tag{1}$$

and

$$\begin{aligned} \min_{\rho^2,\eta_j} \quad & R^2-\rho^2+\frac{1}{\nu_2 l^-}\sum_{j\in I^-}\eta_j \\ s.t. \quad & \|\phi(\mathbf{x}_j)-C\|^2 \geq R^2+\rho^2-\eta_j, \\ & \eta_j \geq 0,\ j\in I^-, \end{aligned} \tag{2}$$

where ν, ν_1 and ν_2 are parameters chosen a priori. C and R are the center and the radius of the small sphere, respectively. $\sqrt{R^2+\rho^2}$ is the radius of the large sphere.

The decision function of MMTSSVM is shown as follows,

$$f(\mathbf{x}) = \begin{cases} 1, & if \|\phi(\mathbf{x})-C\| < \frac{R+\sqrt{R^2+\rho^2}}{2} \\ -1, & else. \end{cases} \tag{3}$$

2.2 Ramp loss SVM

Classic SVM employs the Hinge loss function, which is denoted as follows, $H_s(z) = max(0, s-z)$, where s indicates the position of the Hinge point, to penalize examples classified with an insufficient margin. In terms of the Hinge loss function, the objective function can be written as follows,

$$\min_{\mathbf{w},b} \quad \frac{1}{2}\|\mathbf{w}\|^2 + c\sum_{i=1}^{l} H_1\left(y_i f(\mathbf{x}_i)\right), \tag{4}$$

where the $f(\mathbf{x})$ is the decision function and the expression of $f(\mathbf{x})$ is $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}+b$.

From Fig. 1, we can see that the outliers tend to have the largest loss values according to the Hinge loss function. Therefore, the outliers have a negative influence on constructing the hyperplane and the model is sensitive to outliers. To improve this problem, researchers proposed the Ramp loss function, i.e., the Robust Hinge loss. The expression of the Ramp loss function is shown as follows,

$$R_s(z) = \begin{cases} 0, & z > 1 \\ 1-z, & s \leq z \leq 1 \\ 1-s, & z < s, \end{cases} \tag{5}$$

where $s < 1$ is given a prior. We can see that the Ramp loss function can be expressed by a convex Hinge loss and a concave loss, i.e., $R_s(z) = H_1(z) - H_s(z)$ in Fig. 1. After introducing the Ramp loss function, RSVM is denoted as follows,

$$\begin{aligned} &\min_{\mathbf{w},b} \quad \frac{1}{2}\|\mathbf{w}\|^2 + c\sum_{i=1}^{l} R_s\left(y_i f(\mathbf{x}_i)\right) \\ &= \frac{1}{2}\|\mathbf{w}\|^2 + c\sum_{i=1}^{l} H_1\left(y_i f(\mathbf{x}_i)\right) - c\sum_{i=1}^{l} H_s\left(y_i f(\mathbf{x}_i)\right). \end{aligned} \tag{6}$$

This is a non-differentiable non-convex problem, which is solved by CCCP approach.

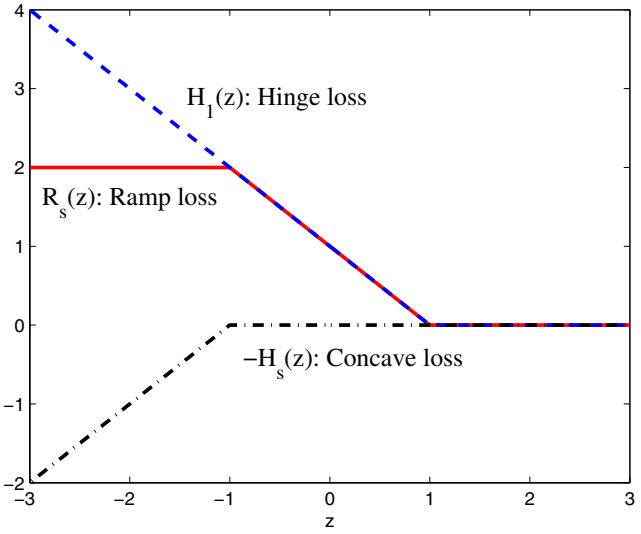

Fig. 1 When s=-1, the geometric decomposition diagram of the Ramp loss function

3 Ramp loss maximum margin of twin spheres support vector machine

Because MMTSSVM is sensitive to outliers, we substitute the Hinge loss function with the Ramp loss function to improve this problem.

We set $u_1 = R^2 - \|\phi(\mathbf{x}_i)-C\|^2$, $u_2 = \|\phi(\mathbf{x}_j)-C\|^2 - (R^2+\rho^2)$. Since the Hinge loss function is $H_s(z) = \max(0, s-z)$, the objective functions of problems (1) and (2) are equal to

$$\min_{R^2,C} \quad R^2 - \frac{\nu}{l^-}\sum_{j\in I^-}\|\phi(\mathbf{x}_j)-C\|^2 + \frac{1}{\nu_1 l^+}H(u_1), \tag{7}$$

and

$$\min_{\rho^2} \quad R^2-\rho^2+\frac{1}{\nu_2 l^-}H(u_2). \tag{8}$$

3.1 Primal Formulations

After replacing the Hinge loss function with the Ramp loss function, we get the primal formulations as follows,

$$\min_{R^2,C} \quad R^2 - \frac{\nu}{l^-}\sum_{j\in I^-}\|\phi(\mathbf{x}_j)-C\|^2 + \frac{1}{\nu_1 l^+}Q(u_1), \tag{9}$$

and

$$\min_{\rho^2} \quad R^2-\rho^2+\frac{1}{\nu_2 l^-}Q(u_2), \tag{10}$$

where $Q(u_1)$ and $Q(u_2)$ are the Ramp loss functions. $Q(u_1)$ and $Q(u_2)$ can be denoted as follows,

$$Q(u_1) = \begin{cases} 0, & u_1 \geq 0 \\ -u_1, & -k_1R^2 < u_1 < 0 \\ k_1R^2, & u_1 \leq -k_1R^2, \end{cases} \tag{11}$$

$$Q(u_2) = \begin{cases} 0, & u_2 \geq 0 \\ -u_2, & -k_2(R^2+\rho^2) < u_2 < 0 \\ k_2(R^2+\rho^2), & u_2 \leq -k_2(R^2+\rho^2), \end{cases} \tag{12}$$

where k_1 and k_2 are chosen a prior.

Then, the problems (9) and (10) can be rewritten as

$$\min_{R^2,C} \quad R^2 - \frac{\nu}{l^-}\sum_{j\in I^-}\|\phi(\mathbf{x}_j) - C\|^2 + \frac{1}{\nu_1 l^+} Q\left(R^2 - \|\phi(\mathbf{x}_i) - C\|^2\right), \tag{13}$$

and

$$\min_{\rho^2} \quad R^2 - \rho^2 + \frac{1}{\nu_2 l^-} Q\left(\|\phi(\mathbf{x}_j) - C\|^2 - (R^2+\rho^2)\right). \tag{14}$$

From (11) and (12), we can see that the Ramp loss function limits the maximum loss value, which reduces the influence of outliers on the optimal solution to the problem. Therefore, the model is less sensitive to outliers.

3.2 Dual problems

The Ramp loss can be decomposed into the sum of a convex Hinge loss and a concave function, i.e.,

$$Q(u_i) = H_1(u_i) - H_2(u_i), \tag{15}$$

where $H_1(u_i) = \max(-u_i, 0)(i = 1, 2)$, $H_2(u_1) = \max(-u_1 - k_1R^2, 0)$, $H_2(u_2) = \max\left(-u_2 - k_2(R^2+\rho^2), 0\right)$.

The (13) and (14) can be rewritten as follows,

$$\min_{R^2,C} \quad \underbrace{R^2 - \frac{\nu}{l^-}\sum_{j\in I^-}\|\phi(\mathbf{x}_j) - C\|^2 + \frac{1}{\nu_1 l^+}H_1(u_1)}_{J_{vex}} \underbrace{-\frac{1}{\nu_1 l^+}H_2(u_1)}_{J_{cav}}, \tag{16}$$

and

$$\min_{\rho^2} \quad \underbrace{R^2 - \rho^2 + \frac{1}{\nu_2 l^-}H_1(u_2)}_{J_{vex}} \underbrace{-\frac{1}{\nu_2 l^-}H_2(u_2)}_{J_{cav}}. \tag{17}$$

We can see that QPP (16) and LPP (17) are both composed of a convex function and a concave function, which can not be solved by quadratic programming. For this reason, we take advantage of CCCP approach to solve this problem for its simple tuning and iteration manner.

We first take QPP (16) into consideration. The objective function is the sum of a convex function $u(\mathbf{x})$ and a concave function $v(\mathbf{x})$. CCCP solves problems by iterating a series of concave functions, i.e.,

$$\mathbf{x}^{t+1} = argmin_{\mathbf{x}} u(\mathbf{x}) + \mathbf{x}^T \nabla v(\mathbf{x}^t), \tag{18}$$

where t means the number of iterations. The convex part of QPP (16) is shown as follows,

$$J_{vex}(C, R^2) = R^2 - \frac{\nu}{l^-}\sum_{j\in I^-}\|\phi(\mathbf{x}_j) - C\|^2 + \frac{1}{\nu_1 l^+}H_1(u_1), \tag{19}$$

and the concave part is shown as follows,

$$J_{cav} = -\frac{1}{\nu_1 l^+}H_2(u_1). \tag{20}$$

The CCCP process for this issue is as follows:

Algorithm 1 CCCP for problem (16)

1: Initialize (C^0, R^0), and set $k = 0$;
2: Solve the problem (21), and obtain the solution (C^{k+1}, R^{k+1});

$$\min_{C,R^2} \quad J_{vex}(C, R^2) + J'_{cav}(R^2 - \|\phi(\mathbf{x}_i) - C\|^2) \cdot (R^2 - \|\phi(\mathbf{x}_i) - C\|^2). \tag{21}$$

3: If $(C^{k+1}, R^{k+1}) \neq (C^k, R^k)$, then set $k = k + 1$ and return to the previous step;
4: If $(C^{k+1}, R^{k+1}) = (C^k, R^k)$, stop the iteration.

Then we rewrite the problem (21) as follows:

$$\begin{aligned} \min_{C,R^2,\xi_i} \quad & R^2 - \frac{\nu}{l^-}\sum_{j\in I^-}\|\phi(\mathbf{x}_j) - C\|^2 + \frac{1}{\nu_1 l^+}\sum_{i\in I^+}\xi_i \\ & + J'_{cav}\left(R^2 - \|\phi(\mathbf{x}_i) - C\|^2\right)\cdot\left(R^2 - \|\phi(\mathbf{x}_i) - C\|^2\right) \\ s.t. \quad & \|\phi(\mathbf{x}_i) - C\|^2 \leq R^2 + \xi_i, \\ & \xi_i \geq 0, i \in I^+. \end{aligned} \tag{22}$$

To simplify the process, we introduce the following symbols,

$$\begin{aligned} \theta_i &= -\frac{1}{\nu_1 l^+}\frac{\partial H_2(u_1)}{\partial u_1} \\ &= \begin{cases} \frac{1}{\nu_1 l^+}, & if\ \|\phi(\mathbf{x}_i) - C\|^2 > R^2 + k_1R^2 \\ 0, & else. \end{cases} \end{aligned} \tag{23}$$

Therefore, problem (22) can be rewritten as follows,

$$\begin{aligned}\min_{C,R^2,\xi_i}\quad & R^2-\frac{\nu}{l^-}\sum_{j\in I^-}\|\phi(\mathbf{x}_j)-C\|^2+\frac{1}{\nu_1 l^+}\sum_{i\in I^+}\xi_i\\ & +\sum_{i\in I^+}\theta_i\left(R^2-\|\phi(\mathbf{x}_i)-C\|^2\right)\\ s.t.\quad & \|\phi(\mathbf{x}_i)-C\|^2\le R^2+\xi_i,\\ & \xi_i\ge 0, i\in I^+.\end{aligned}\tag{24}$$

Similarly, the CCCP structure of problem (17) can also be given. The convex part of this problem is shown as follows:

$$J_{vex}(\rho^2)=R^2-\rho^2+\frac{1}{\nu_2 l^-}H_1(u_2),\tag{25}$$

and the concave part is shown as follows:

$$J_{cav}(\rho^2)=-\frac{1}{\nu_2 l^-}H_2(u_2).\tag{26}$$

The CCCP process for problem (17) is as follows:

Algorithm 2 CCCP for problem (17)

1: Initialize ρ^0, and set $k=0$;
2: Solve the problem (27), and obtain the solution ρ^{k+1};

$$\begin{aligned}\min_{\rho^2}\quad & J_{vex}(\rho^2)+J'_{cav}\left(\|\phi(\mathbf{x}_j)-C\|^2-(R^2+\rho^2)\right)\\ & \cdot\left(\|\phi(\mathbf{x}_j)-C\|^2-(R^2+\rho^2)\right).\end{aligned}\tag{27}$$

3: If $\rho^{k+1}\neq\rho^k$, then set $k=k+1$ and return to the previous step;
4: If $\rho^{k+1}=\rho^k$, stop the iteration.

Then we rewrite the problem (27) as follows:

$$\begin{aligned}\min_{\rho^2,\eta_j}\quad & R^2-\rho^2+\frac{1}{\nu_2 l^-}\sum_{j\in I^-}\eta_j+J'_{cav}\left(\|\phi(\mathbf{x}_j)-C\|^2\right.\\ & \left.-\left(R^2+\rho^2\right)\right)\left(\|\phi(\mathbf{x}_j)-C\|^2-\left(R^2+\rho^2\right)\right)\\ s.t.\quad & \|\phi(\mathbf{x}_j)-C\|^2\ge R^2+\rho^2-\eta_j,\\ & \eta_j\ge 0, j\in I^-.\end{aligned}\tag{28}$$

To simplify the process, we introduce the following symbols,

$$\begin{aligned}\delta_j &= -\frac{1}{\nu_2 l^-}\frac{\partial H_2(u_2)}{\partial u_2}\\ &=\begin{cases}\frac{1}{\nu_2 l^-}, & if\ \|\phi(\mathbf{x}_j)-C\|^2<R^2+\rho^2\\ -k_2\left(R^2+\rho^2\right) & \\ 0, & else.\end{cases}\end{aligned}\tag{29}$$

Therefore, problem (28) can be rewritten as follows,

$$\begin{aligned}\min_{\rho^2,\eta_j}\quad & R^2-\rho^2+\frac{1}{\nu_2 l^-}\sum_{j\in I^-}\eta_j\\ & +\sum_{j\in I^-}\delta_j\left(\|\phi\left(\mathbf{x}_j\right)-C\|^2-\left(R^2+\rho^2\right)\right)\\ s.t.\quad & \|\phi(\mathbf{x}_j)-C\|^2\ge R^2+\rho^2-\eta_j,\\ & \eta_j\ge 0, j\in I^-.\end{aligned}\tag{30}$$

To solve the problem (24), we introduce the Lagrange function which is shown as follows,

$$\begin{aligned}L_1 = & R^2-\frac{\nu}{l^-}\sum_{j\in I^-}\|\phi(\mathbf{x}_j)-C\|^2+\frac{1}{\nu_1 l^+}\sum_{i\in I^+}\xi_i\\ & +\sum_{i\in I^+}\theta_i\left(R^2-\|\phi(\mathbf{x}_i)-C\|^2\right)\\ & +\sum_{i\in I^+}\alpha_i\left(\|\phi(\mathbf{x}_i)-C\|^2-R^2-\xi_i\right)-\sum_{i\in I^+}\beta_i\xi_i.\end{aligned}\tag{31}$$

where $\alpha_i\ge 0$, $\beta_i\ge 0$ both are Lagrange multipliers. Differentiating the Lagrangian function L_1 with respect to variables R^2, C and ξ_i yields the following Karush-Kuhn-Tucker(KKT) conditions:

$$\frac{\partial L_1}{\partial R^2}=1+\sum_{i\in I^+}\theta_i-\sum_{i\in I^+}\alpha_i=0,\tag{32}$$

$$\begin{aligned}\frac{\partial L_1}{\partial C} = & \frac{2\nu}{l^-}\sum_{j\in I^-}\left(\phi(\mathbf{x}_j)-C\right)+2\sum_{i\in I^+}\theta_i\left(\phi(\mathbf{x}_i)\right.\\ & \left.-C\right)-2\sum_{i\in I^+}\alpha_i\left(\phi(\mathbf{x}_i)-C\right)=0,\end{aligned}\tag{33}$$

$$\frac{\partial L_1}{\partial \xi_i}=\frac{1}{\nu_1 l^+}-\alpha_i-\beta_i=0.\tag{34}$$

From (32), we can get

$$\sum_{i\in I^+}(\alpha_i-\theta_i)=1.\tag{35}$$

From (33), we can obtain the center C as follows:

$$C=\frac{1}{1-\nu}\left(\sum_{i\in I^+}(\alpha_i-\theta_i)\phi(\mathbf{x}_i)-\frac{\nu}{l^-}\sum_{j\in I^-}\phi(\mathbf{x}_j)\right).\tag{36}$$

Then

$$< C, C > = \frac{1}{(1-\nu)^2}\left(\sum_{i\in I^+}\sum_{j\in I^+}(\alpha_i-\theta_i) \cdot(\alpha_j-\theta_j)K(\mathbf{x}_i,\mathbf{x}_j) + \left(\frac{\nu}{l^-}\right)^2\sum_{i\in I^-}\sum_{j\in I^-}K(\mathbf{x}_i,\mathbf{x}_j) - \frac{2\nu}{l^-}\sum_{i\in I^+}\sum_{j\in I^-}(\alpha_i-\theta_i)K(\mathbf{x}_i,\mathbf{x}_j)\right). \quad (37)$$

Finally, we derive the dual formulation as follows:

$$\begin{aligned}\max_{\boldsymbol{\alpha}}\ & -\sum_{i\in I^+}\sum_{j\in I^+}(\alpha_i-\theta_i)(\alpha_j-\theta_j)K(\mathbf{x}_i,\mathbf{x}_j) \\ & +\sum_{i\in I^+}(\alpha_i-\theta_i)\left(\frac{2\nu}{l^-}\sum_{j\in I^-}K(\mathbf{x}_j,\mathbf{x}_i) \right. \\ & \left. +(1-\nu)K(\mathbf{x}_i,\mathbf{x}_i)\right) \\ s.t.\ & 0\le\alpha_i\le\frac{1}{\nu_1 l^+},\sum_{i\in I^+}(\alpha_i-\theta_i)=1. \end{aligned} \quad (38)$$

In order to obtain the radius of the small sphere, we use positive points $\mathbf{x}_i$ whose Lagrangian multiplier α_i satisfy $0 < \alpha_i < \frac{1}{\nu_1 l^+}$. Supposing the number of $\mathbf{x}_i$ above is n_p. According to KKT condition, we can get: $\alpha_i\left(\|\phi(\mathbf{x}_i)-C\|^2-R^2-\xi_i\right)=0$. Then we can acquire the square radius of the small sphere as

$$R^2=\frac{1}{n_p}\sum_{i=1}^{n_p}\|\phi(\mathbf{x}_i)-C\|^2=\frac{1}{n_p}\sum_{i=1}^{n_p}\left(K(\mathbf{x}_i,\mathbf{x}_i)+C^2 -\frac{2}{1-\nu}\left(\sum_{j\in I^+}(\alpha_j-\theta_j)K(\mathbf{x}_i,\mathbf{x}_j)-\frac{\nu}{l^-}\sum_{j\in I^-}K(\mathbf{x}_i,\mathbf{x}_j)\right)\right). \quad (39)$$

To solve the problem (30), we introduce the Lagrange function which is shown as follows,

$$L_2 = R^2-\rho^2+\frac{1}{\nu_2 l^-}\sum_{j\in I^-}\eta_j+\sum_{j\in I^-}\delta_j\left(\|\phi(\mathbf{x}_j)-C\|^2-(R^2+\rho^2)\right)-\sum_{j\in I^-}\gamma_j\left(\|\phi(\mathbf{x}_j)-C\|^2-R^2-\rho^2+\eta_j\right)-\sum_{j\in I^-}\lambda_j\eta_j. \quad (40)$$

where $\gamma_j \ge 0$ and $\lambda_j \ge 0$ are Lagrange multipliers. Differentiating the Lagrange function L_2 with respect to variables ρ^2 and η_j yields the following KKT conditions:

$$\frac{\partial L_2}{\partial\rho^2}=-1-\sum_{j\in I^-}\delta_j+\sum_{j\in I^-}\gamma_j=0, \quad (41)$$

$$\frac{\partial L_2}{\partial\eta_j}=\frac{1}{\nu_2 l^-}-\gamma_j-\lambda_j=0. \quad (42)$$

From (41), we can obtain

$$\sum_{j\in I^-}\left(\gamma_j-\delta_j\right)=1. \quad (43)$$

Finally, we derive the dual formulation as follows:

$$\begin{aligned}\max_{\boldsymbol{\gamma}}\ & -\sum_{j\in I^-}\left(\gamma_j-\delta_j\right)\left(\|\phi(\mathbf{x}_j)-C\|^2\right) \\ s.t.\ & \sum_{j\in I^-}(\gamma_j-\delta_j)=1, 0\le\gamma_j\le\frac{1}{\nu_2 l^-}. \end{aligned} \quad (44)$$

In order to obtain ρ^2, we use negative points $\mathbf{x}_j$ whose Lagrangian multiplier γ_j satisfy $0 < \gamma_j < \frac{1}{\nu_2 l^-}$. Supposing the number of $\mathbf{x}_j$ is n_n. According to KKT condition: $\gamma_j(\|\phi(\mathbf{x}_j)-C\|^2-R^2-\rho^2+\eta_j)=0$, we can acquire

$$\rho^2=\frac{1}{n_n}\sum_{j=1}^{n_n}\left(\|\phi\left(\mathbf{x}_j\right)-C\|^2-R^2\right). \quad (45)$$

Then, CCCP procedure based on Ramp-MMTSSVM is summarized as follows.

Algorithm 3 Ramp-MMTSSVM

1: Initialize $\boldsymbol{\theta}^0=\mathbf{0}$ and $\boldsymbol{\delta}^0=\mathbf{0}$.
2: Construct and deal with the dual problem (38) and (44) in the kth iteration step to get the optimal solution $\boldsymbol{\alpha}^k$ and $\boldsymbol{\gamma}^k$, then calculate C^k, $(R^2)^k$ and $(\rho^2)^k$ according to (36), (39) and (45), respectively. Construct the functions $\|\phi(\mathbf{x}_i)-C^k\|^2(i\in I^+)$ and $\|\phi(\mathbf{x}_j)-C^k\|^2(j\in I^-)$.
3: Update $\boldsymbol{\theta}^k$ and $\boldsymbol{\delta}^k$ according to (23) and (29).
4: If $\boldsymbol{\theta}^{k+1}=\boldsymbol{\theta}^k$ and $\boldsymbol{\delta}^{k+1}=\boldsymbol{\delta}^k$, stop the iteration and get the optimal solutions $C=C^k$, $R^2=(R^2)^k$ and $\rho^2=(\rho^2)^k$. Otherwise, set $k=k+1$, and go to the step 2.

3.3 Property of parameters ν_1 and ν_2

In the Ramp-MMTSSVM, parameters ν_1 and ν_2 have their theoretical significance. In the following part, we will analyze the properties of ν_1 and ν_2.

To make it easier to understand, we first give the following two definitions.

Definition 1 The small sphere can be divided into four sets $S_+=\{i|\|\phi(\mathbf{x}_i)-C\|^2<R^2\}$, $B_+=\{i|\|\phi(\mathbf{x}_i)-C\|^2=$

$R^2\}$, $R_+ = \{i|R^2 < \|\phi(\mathbf{x}_i) - C\|^2 < R^2 + k_1R^2\}$ and $W_+ = \{i|\|\phi(\mathbf{x}_i) - C\|^2 \geq R^2 + k_1R^2\}$.

Definition 2 The large sphere can be divided into four sets $S_- = \{j|\|\phi(\mathbf{x}_j) - C\|^2 \leq R^2 + \rho^2 - k_2(R^2 + \rho^2)\}$, $B_- = \{j|R^2 + \rho^2 - k_2(R^2 + \rho^2) < \|\phi(\mathbf{x}_j) - C\|^2 < R^2 + \rho^2\}$, $R_- = \{j|\|\phi(\mathbf{x}_j) - C\|^2 = R^2 + \rho^2\}$ and $W_- = \{j|\|\phi(\mathbf{x}_j) - C\|^2 > R^2 + \rho^2\}$.

According to the two definitions above, we can obtain the following propositions.

Proposition 1 *Let n_1 represent the number of positive samples in B_+, R_+ and W_+, n_2 represent the number of positive samples in R_+ and W_+, n_3 represent the number of positive samples in W_+. Thus, we can obtain $\frac{n_2-n_3}{l^+} \leq \nu_1 \leq \frac{n_1-n_3}{l^+}$.*

Proof If $\mathbf{x}_i (i = 1, 2, \ldots, l^+)$ represents a positive data point in S_+, we can get $\xi_i = 0$. Then, according to KKT condition $\alpha_i(\|\phi(\mathbf{x}_i) - C\|^2 - R^2 - \xi_i) = 0$, we can get $\alpha_i = 0$. From (23), we can acquire $\theta_i = 0$. If $\mathbf{x}_i$ represents a positive data point in B_+, we can get $\xi_i = 0$, then according to KKT condition $\beta_i\xi_i = 0$, we can get $\beta_i \geq 0$. Therefore, $0 \leq \alpha_i \leq \frac{1}{\nu_1 l^+}$, and $\theta_i = 0$. If $\mathbf{x}_i$ represents a positive data point in R_+, we can get $\xi_i \neq 0$, then $\beta_i = 0$, then $\alpha_i = \frac{1}{\nu_1 l^+}$ and $\theta_i = 0$. If $\mathbf{x}_i$ represents a positive data point in W_+, we can get $\alpha_i = \frac{1}{\nu_1 l^+}$ and $\theta_i = \frac{1}{\nu_1 l^+}$. According to the conclusions above, we have $\frac{n_2}{\nu_1 l^+} \leq \sum_{i\in I^+} \alpha_i \leq \frac{n_1}{\nu_1 l^+}$ and $-\sum_{i\in I^+} \theta_i = -\frac{n_3}{\nu_1 l^+}$. Therefore, $\frac{n_2-n_3}{\nu_1 l^+} \leq \sum_{i\in I^+}(\alpha_i - \theta_i) \leq \frac{n_1-n_3}{\nu_1 l^+}$. From (35), we can get $\frac{n_2-n_3}{\nu_1 l^+} \leq 1 \leq \frac{n_1-n_3}{\nu_1 l^+}$. Then, $\frac{n_2-n_3}{l^+} \leq \nu_1 \leq \frac{n_1-n_3}{l^+}$. □

Proposition 2 *Let m_1 represent the number of negative samples in S_-, B_- and R_-, m_2 represent the number of negative samples in S_- and B_-, m_3 represent the number of negative samples in S_-. Thus, we can obtain $\frac{m_2-m_3}{l^-} \leq \nu_2 \leq \frac{m_1-m_3}{l^-}$.*

Proof If $\mathbf{x}_j (i = 1, 2, \ldots, l^-)$ represents a negative data point in S_-, we can get $\eta_j \neq 0$, then according to KKT condition $\lambda_j\eta_j = 0$, we can get $\lambda_j = 0$. From (42), we can acquire $\gamma_j = \frac{1}{\nu_2 l^-}$. And from (29), we can get $\delta_j = \frac{1}{\nu_2 l^-}$. If $\mathbf{x}_j$ represents a negative data point in B_-, we can get $\gamma_j = \frac{1}{\nu_2 l^-}$, and $\delta_j = 0$. If $\mathbf{x}_j$ represents a negative data point in R_-, we can get $\eta_j = 0$, then $\lambda_j \geq 0$, then $0 \leq \gamma_j \leq \frac{1}{\nu_2 l^-}$ and $\delta_j = 0$. If $\mathbf{x}_j$ represents a negative data point in W_-, we can get $\eta_j = 0$, then $\gamma_j = 0$ and $\delta_j = 0$. According to conclusions above, we have $\frac{m_2}{\nu_2 l^-} \leq \sum_{j\in I^-} \gamma_j \leq \frac{m_1}{\nu_2 l^-}$ and $-\sum_{j\in I^-} \delta_j = -\frac{m_3}{\nu_2 l^-}$. Therefore, $\frac{m_2-m_3}{\nu_2 l^-} \leq \sum_{j\in I^-}(\gamma_j - \delta_j) \leq \frac{m_1-m_3}{\nu_2 l^-}$. From (43), we can get $\frac{m_2-m_3}{\nu_2 l^-} \leq 1 \leq \frac{m_1-m_3}{\nu_2 l^-}$. Then, $\frac{m_2-m_3}{l^-} \leq \nu_2 \leq \frac{m_1-m_3}{l^-}$. □

From Propositions above, we can acquire the range of ν_1 and ν_2: $0 < \nu_1, \nu_2 < 1$.

4 Multicategory Ramp loss maximum margin of twin spheres support vector machine

Because multi-class classification problems are more common, we extend Ramp-MMTSSVM to multi-class classification problems by RVO strategy. For a K-class classification problem, we choose the ith class as the negative class and the rest classes as the positive class to generate a Ramp-MMTSSVM. Thus, K classifiers will finally be generated. The optimization problems of MRMMTSSVM are shown as follows:

$$\begin{aligned}\min_{C_k, R_k^2, \xi_i} \quad & R_k^2 - \frac{\nu_k}{l_k}\sum_{j\in A_k}\|\phi(\mathbf{x}_j) - C_k\|^2 + \frac{1}{\nu_{1k} l_{K-1}}\sum_{i\in B_k}\xi_i \\ & + \sum_{i\in B_k}\theta_i\left(R_k^2 - \|\phi(\mathbf{x}_i) - C_k\|^2\right) \\ s.t. \quad & \|\phi(\mathbf{x}_i) - C_k\|^2 \leq R_k^2 + \xi_i, \\ & \xi_i \geq 0, i \in B_k. \end{aligned} \tag{46}$$

and

$$\begin{aligned}\min_{\rho_k^2, \eta_j} \quad & R_k^2 - \rho_k^2 + \frac{1}{\nu_{2k} l_k}\sum_{j\in A_k}\eta_j \\ & + \sum_{j\in A_k}\delta_j\left(\|\phi\left(\mathbf{x}_j\right) - C_k\|^2 - \left(R_k^2 + \rho_k^2\right)\right) \\ s.t. \quad & \|\phi\left(\mathbf{x}_j\right) - C_k\|^2 \geq R_k^2 + \rho_k^2 - \eta_j, \\ & \eta_j \geq 0, j \in A_k. \end{aligned} \tag{47}$$

where l_k denotes the number of data points of kth class and l_{K-1} denotes the number of data points of the rest classes, $A_k \in R^{l_k \times n}(k = 1, \ldots, K)$, and $B_k = [A_1^T, \ldots A_{k-1}^T, A_{k+1}^T, \ldots A_K^T]^T$.

For solving each classifier, we also use CCCP approach. The usage of CCCP approach is the same as the binary classification case. Finally, we adopt 'voting' scheme to obtain the class of each new data point.

5 Numerical experiments

In this section, we conduct experiments on one artificial dataset and twenty benchmark datasets to demonstrate the validity of the algorithms we proposed.

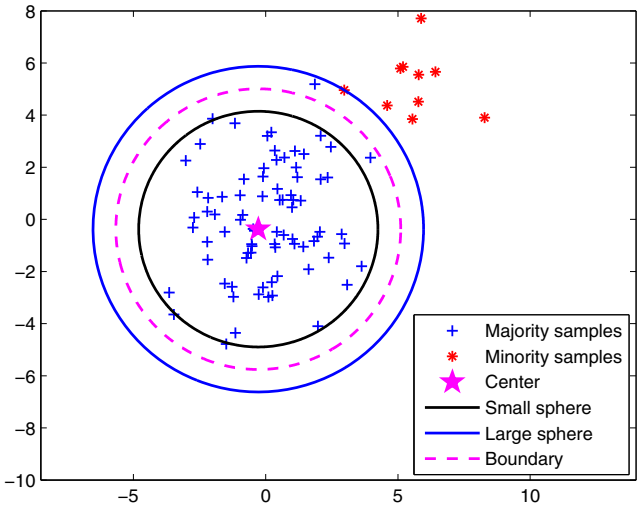

Fig. 2 Illustration of linear Ramp-MMTSSVM

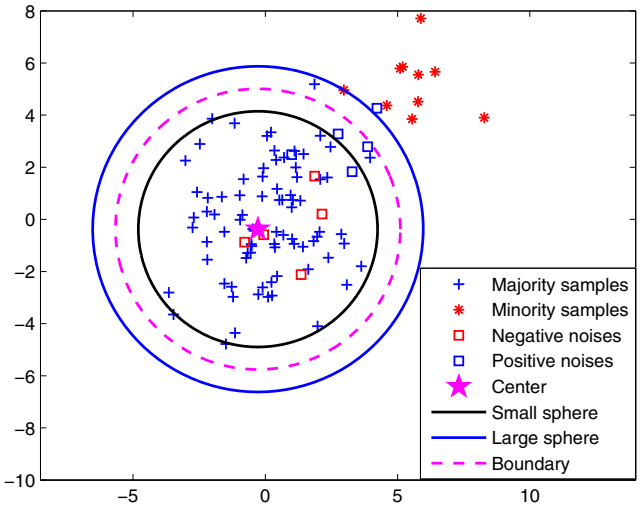

Fig. 4 Illustration of linear Ramp-MMTSSVM when handling the noise data points

5.1 Experiments on one artificial dataset

We generate a 2-D artificial dataset, including 80 positive data points and 10 negative data points. The positive data points follow uniform distribution N(0,0,3,3) and the negative data points follow N(3,3,2,2). The illustration of linear Ramp-MMTSSVM on this dataset is shown in Fig. 2.

In this part, we compare linear Ramp-MMTSSVM and linear MMTSSVM which are shown in Figs. 2 and 3 on the artificial dataset. From the pictures, it is easy to find that the small sphere captures more positive data points in Ramp-MMTSSVM than that in MMTSSVM. This represents Ramp-MMTSSVM has better experimental performance than MMTSSVM on this artificial dataset.

To testify that the Ramp loss can reduce the effect of outliers, we add five noise points to the positive and negative class respectively. After adding noises, the performance of Ramp-MMTSSVM and MMTSSVM are shown in Figs. 4 and 5 respectively. From Fig. 5, it is obvious that the boundary of MMTSSVM expands outward. However, the boundary of Ramp-MMTSSVM in Fig. 4 has almost no change. That is to say, when handling noises and outliers, Ramp-MMTSSVM has better performance than MMTSSVM.

In addition, in order to better show the property of ν_1 which is mentioned in part 3, we depict Fig. 6, where the $x-axis$ represents the values of parameter ν_1. In the picture, the black dotted line indicates when ν_1 changes,

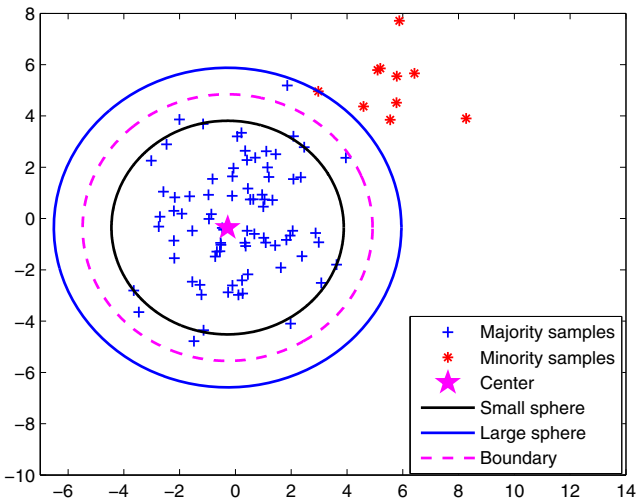

Fig. 3 Illustration of linear MMTSSVM

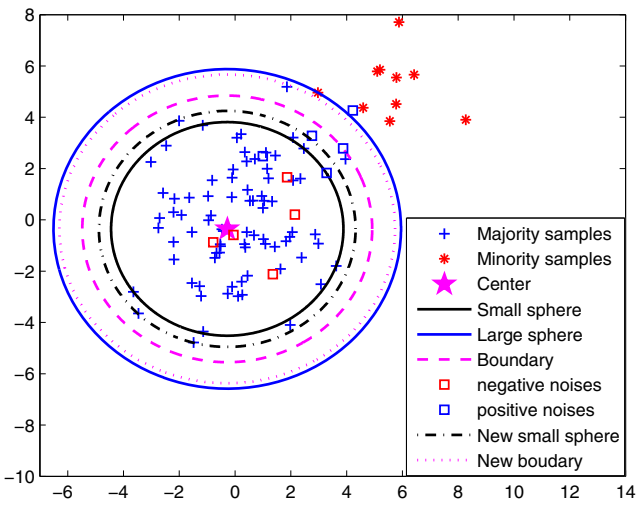

Fig. 5 Illustration of linear MMTSSVM when handling the noise data points

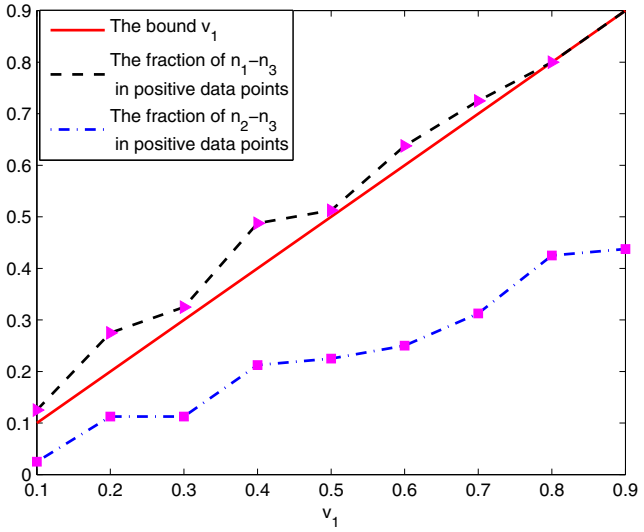

Fig. 6 The property of parameter ν_1

the changing proportion of $n_1 - n_3$ in positive data points. The red solid line represents the changing values of ν_1. The blue dotted line indicates when ν_1 changes, the changing proportion of $n_2 - n_3$ in positive data points.

From the picture, we can get that parameter ν_1 is a lower bound on the fraction of $n_1 - n_3$ in the positive data points and an upper bound on the fraction of $n_2 - n_3$ in the positive data points, which is helpful for us to select the range of parameters.

5.2 Experiments on benchmark datasets

In this part, we make experiments on twenty benchmark datasets, which are collected from UCI machine learning repository. The detailed information of twenty datasets is shown in Table 1.

In the binary classification case, Ramp-MMTSSVM is compared with THSVM, SSLM and MMTSSVM. In the multi-class classification case, MRMMTSSVM is compared with OVO-THSVM, THKSVM and OVR-MMTSSVM.

We adopt 5-fold cross-validation for each experiment to make the experiments more convincing. All experiments are carried out in Matlab R2014a on Windows 7 running on a PC with system configuration Inter Core i3-4160Duo CPU(3.60GHz)with 4.00 GB of RAM.

5.2.1 Parameters selection

The approaches we mentioned above, i.e., THSVM, SSLM, MMTSSVM, THKSVM, Ramp-MMTSSVM and MRMMTSSVM all depend heavily on parameters selection. In these experiments, we choose the Gaussian kernel function $k(\mathbf{x}_i, \mathbf{x}_j) = exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/\sigma^2)$. We acquire optimal values of the parameters by the grid search method. All the Gaussian kernel parameter σ is selected from the set $\{2^i | i = -3, -1, \ldots, 7\}$.

In THSVM, for reducing computational complexity, we set $c_1 = c_2$ and $\nu_1 = \nu_2$, which are chosen from the set $\{2^i | i = 0, 2, \ldots, 8\}$ and $\{0.1, 0.2, \ldots, 0.9\}$, respectively.

In SSLM, we set $\nu_1 = \nu_2$, which are selected from the set $\{0.001, 0.01\}$ and the range of ν is $\{0.1, 0.2, \ldots, 0.9\}$.

In MMTSSVM, we set $\nu_1 = \nu_2$. All the ν are selected from the set $\{0.1, 0.2, \ldots, 0.9\}$.

In THKSVM, the range of ν_k is $\{2^i | i = -4, -2, \ldots, 4\}$. And d_k is selected from the set $\{2^i | i = 1, 3, \ldots, 9\}$.

In our approach, we set $k_1 = k_2$ whose range is $\{0.5, 1, 1.5, 2\}$. The range of other parameters in our approaches is the same as that in MMTSSVM.

Table 1 The characteristics of twenty benchmark datasets

Datasets	♯ Examples	♯ Features	♯ Classes
Sonar	208	60	2
Breast cancer	569	30	2
Ionosphere	351	34	2
Heart	294	13	2
Liver disorder	345	6	2
Pima	768	8	2
Spectf heart	267	44	2
Banknote	1372	4	2
Monks	432	6	2
Abalone	2835	8	2
Iris	150	4	3
Soybean	47	35	4
Seeds	210	7	3
Ecoli	327	7	5
Optidigits	2268	64	4
Hayes-roth	132	5	3
Teaching	151	5	3
Balance	625	4	3
Image	2310	19	7
Cmc	1473	10	3

5.2.2 Result analysis

In Tables 2 and 3, 'Accuracy' means the average value of testing results and plus or minus the corresponding standard deviation. 'Time' means the mean value of the time, including training time and testing time.

From Table 2, we can see in binary classification case, the experimental results of our approach are better than other algorithms on most datasets, i.e., Sonar, Breast cancer, Heart, Pima, Spectf heart and Abalone dataset. MMTSSVM obtains the best performance on Ionosphere, Banknote and

Table 2 The experimental results for binary classification case on ten benchmark datasets

Datasets	THSVM		SSLM		MMTSSVM		Ramp-MMTSSVM	
	Accuracy(%) (c_1, ν_1, r)	Time(s)	Accuracy(%) (ν_1, ν, r)	Time(s)	Accuracy(%) (ν_1, ν, r)	Time(s)	Accuracy(%) (ν_1, k_1, ν, r)	Time(s)
Sonar	66.98±11.79 (16,0.9,128)	0.05	61.40±12.80 (0.001,0.2,128)	0.05	70.23±6.86 (0.6,0.8,8)	0.04	**72.09±11.74** (0.6,0.5,0.8,32)	0.04
Breast cancer	71.83±7.52 (4,0.9,32)	0.26	91.65±2.27 (0.001,0.7,128)	0.59	92.70±2.51 (0.3,0.3,32)	0.27	**93.22±1.89** (0.2,2,0.8,128)	0.64
Ionosphere	81.69±13.66 (64,0.5,0.5)	0.11	86.20±6.25 (0.01,0.9,8)	0.18	**92.96±3.30** (0.2,0.9,2)	0.13	91.83±4.61 (0.3,1,0.9,2)	0.14
Heart	67.00±3.21 (16,0.9,8)	0.09	63.67±2.17 (0.01,0.9,128)	0.13	68.00±5.70 (0.4,0.9,128)	0.09	**70.00±5.40** (0.7,0.5,0.9,32)	0.08
Liver disorder	**69.57±6.06** (64,0.9,8)	0.11	57.10±1.65 (0.01,0.5,128)	0.17	65.80±4.98 (0.5,0.9,128)	0.12	67.83±5.74 (0.8,1.5,0.9,8)	0.09
Pima	66.62±1.56 (256,0.4,8)	0.41	70.26±2.40 (0.001,0.4,128)	1.83	72.34±3.27 (0.3,0.8,128)	0.51	**73.25±3.74** (0.2,0.5,0.9,128)	0.51
Spectf heart	82.22±7.00 (4,0.9,32)	0.09	74.81±7.12 (0.01,0.4,128)	0.11	82.22±3.84 (0.6,0.8,32)	0.08	**82.96±6.73** (0.2,0.5,0.8,32)	0.08
Banknote	89.60±7.70 (4,0.7,0.125)	1.43	77.60±2.49 (0.01,0.1,0.125)	3.91	**93.38±1.65** (0.3,0.9,0.125)	1.20	92.65±1.69 (0.3,1.5,0.7,0.125)	1.19
Monks	66.82±20.87 (16,0.3,0.5)	0.13	55.00±8.26 (0.01,0.5,2)	0.26	**71.36±12.92** (0.7,0.9,8)	0.13	69.09±7.29 (0.9,1.5,0.7,2)	0.12
Abalone	96.51±4.82 (64,0.1,0.125)	7.62	89.12±2.71 (0.01,0.1,0.125)	32.46	96.30±1.53 (0.1,0.7,0.5)	7.24	**97.32±1.07** (0.1,0.5,0.8,2)	6.86

Bold type shows the best result

Table 3 The experimental results for multi-class classification case on ten benchmark datasets

Datasets	OVO-THSVM		THKSVM		OVR-MMTSSVM		MRMMTSSVM	
	Accuracy(%) (c_1, ν_1, r)	Time(s)	Accuracy(%) (ν_k, d_k, r)	Time(s)	Accuracy(%) (ν_1, ν, r)	Time(s)	Accuracy(%) $(\nu_{1k}, k_1, \nu_k, r)$	Time(s)
Iris	64.00±5.48 (16,0.6,0.125)	0.06	**96.67±3.33** (16,8,2)	0.05	92.00±2.98 (0.2,0.2,2)	0.08	95.33±3.80 (0.4,1.5,0.7,0.5)	0.07
Soybean	82.00±8.37 (64,0.1,2)	0.03	98.00±4.47 (1,2,2)	0.04	98.00±4.47 (0.1,0.9,2)	0.05	**100.00±0.00** (0.2,0.5,0.9,128)	0.04
Seeds	80.95±10.10 (256,0.5,0.5)	0.08	**89.52±9.46** (16,128,2)	0.09	88.10±7.14 (0.6,0.6,8)	0.11	**89.52±8.00** (0.9,1.5,0.4,2)	0.14
Ecoli	66.87±5.21 (4,0.5,0.125)	0.38	78.81±3.24 (16,128,0.125)	0.30	73.73±6.03 (0.4,0.4,32)	0.36	**85.67±3.09** (0.3,2,0.8,0.125)	0.57
Optidigits	72.94±6.70 (256,0.8,8)	6.97	92.11±1.61 (16,8,32)	20.24	96.32±2.17 (0.2,0.6,32)	19.61	**98.33±1.16** (0.1,2,0.6,32)	33.05
Hayes-roth	47.14±2.99 (4,0.4,2)	0.05	48.57±7.41 (0.25,2,2)	0.04	42.14±6.39 (0.3,0.9,8)	0.06	**57.86±8.89** (0.6,2,0.4,2)	0.06
Teaching	65.16±26.44 (256,0.5,0.5)	0.04	66.45±22.65 (1,512,0.125)	0.05	54.84±33.60 (0.9,0.9,2)	0.07	**67.74±22.81** (0.8,1,0.3,0.125)	0.06
Balance	78.73±3.24 (256,0.1,0.5)	0.43	80.79±8.44 (16,32,2)	0.78	81.11±8.61 (0.3,0.8,2)	0.70	**88.25±4.71** (0.2,1.5,0.7,2)	0.94
Image	68.27±10.20 (64,0.8,8)	9.04	76.62±11.26 (16,512,8)	38.74	80.65±9.31 (0.2,0.6,32)	38.46	**81.17±9.57** (0.2,0.5,0.8,32)	53.52
Cmc	**87.57±2.21** (16,0.7,2)	2.15	59.12±6.56 (16,32,2)	4.26	85.81±3.68 (0.4,0.9,2)	3.73	86.08±2.64 (0.2,1,0.9,2)	5.06

Bold type shows the best result

Table 4 Average ranks of four algorithms in binary classification case

Datasets	THSVM	SSLM	MMTSSVM	Ramp-MMTSSVM
Sonar	3	4	2	1
Breast cancer	4	3	2	1
Ionosphere	4	3	1	2
Heart	3	4	2	1
Liver disorder	1	4	3	2
Pima	4	3	2	1
Spectf heart	2.5	4	2.5	1
Banknote	3	4	1	2
Monks	3	4	1	2
Abalone	2	4	3	1
Average rank	2.95	3.70	1.95	1.40

Monks dataset. However, the accuracy of our algorithm is lower than MMTSSVM within a little range and higher than THSVM and SSLM. THSVM performs the best on Liver disorder dataset. Also, the performance of our approach is slightly worse than THSVM, which ranks the second. In binary classification case, SSLM never acquires the best performance on the ten datasets.

From Table 3, we can see in multi-class classification case, MRMMTSSVM has the best experimental results on most datasets, i.e., Soybean, Ecoli, Optidigits, Hayes-roth, Teaching, Balance and Image dataset. For Seeds dataset, THKSVM and MRMMTSSVM acquire the same accuracy, which is the highest. THKSVM obtains the best performance on Iris dataset, but MRMMTSSVM is a little worse than THKSVM, ranking the second. On Cmc dataset, OVO-THSVM gets the best experimental result. Though the accuracy produced by MRMMTSSVM is not the highest on Cmc dataset, it is better than THKSVM and OVR-MMTSSVM. In multi-class classification case, OVR-MMTSSVM never obtains the highest accuracy on the ten datasets.

In conclusion, we can see that both in binary and multi-class classification case, our algorithms perform better than other algorithms.

5.2.3 Friedman test

In order to better analyse the experimental performance of eight algorithms statically, we introduce Friedman Test. Tables 4 and 5 show the average ranking of four algorithms in binary and multi-class classification case respectively.

Under the null-hypothesis that all the algorithms are equivalent, one can compute the Friedman statistic according to:

$$\chi_F^2 = \frac{12N}{k(k+1)}\left[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right], \tag{48}$$

where $R_j = \frac{1}{N}\sum_i r_i^j$ and r_i^j represents the jth of k algorithms on the ith of N datasets. Friedman's χ_F^2 is undesirably conservative and derives a better statistic

$$F_F = \frac{(N-1)\chi_F^2}{N(k-1)-\chi_F^2}, \tag{49}$$

which is distributed according to the F-distribution with $k-1$ and $(k-1)(N-1)$ degrees of freedom.

According to (48) and (49), we can obtain in binary classification case, $\chi_F^2 = 18.9300$ and $F_F = 15.3902$, where F_F is distributed according to F-distribution with (3,27) degrees of freedom. The critical value of $F(3,27)$ is 2.96 for the level of significance $\alpha = 0.05$, and similarly it is 3.65 for $\alpha = 0.025$ and 4.60 for $\alpha = 0.01$. We can see that the value of F_F is much larger than the critical value which means our approach in binary classification case has significant difference with other algorithms. In addition, from Table 4, it is clear that the average ranking of our approach is far lower than that of the remaining algorithms, which means our approach is more valid than other three algorithms in binary classification case.

Table 5 Average ranks of four algorithms in multi-class classification case

Datasets	OVO-THSVM	THKSVM	OVR-MMTSSVM	MRMMTSSVM
Iris	4	1	3	2
Soybean	4	2.5	2.5	1
Seeds	4	1.5	3	1.5
Ecoli	4	2	3	1
Optidigits	4	3	2	1
Hayes-roth	3	2	4	1
Teaching	3	2	4	1
Balance	4	3	2	1
Image	4	3	2	1
Cmc	1	4	3	2
Average rank	3.50	2.40	2.85	1.25

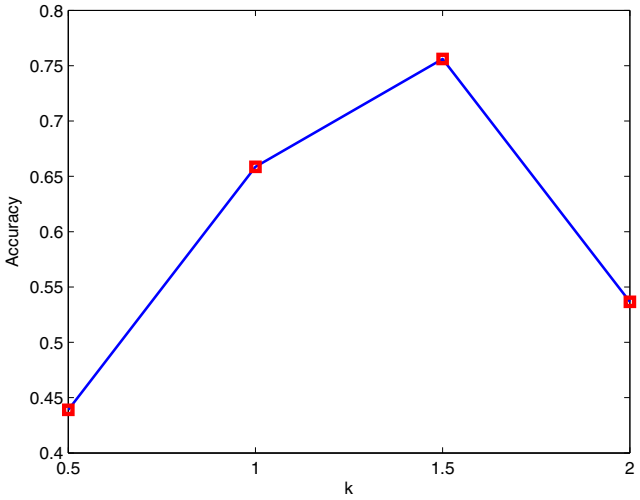

Fig. 7 The effect of parameter k on the performance

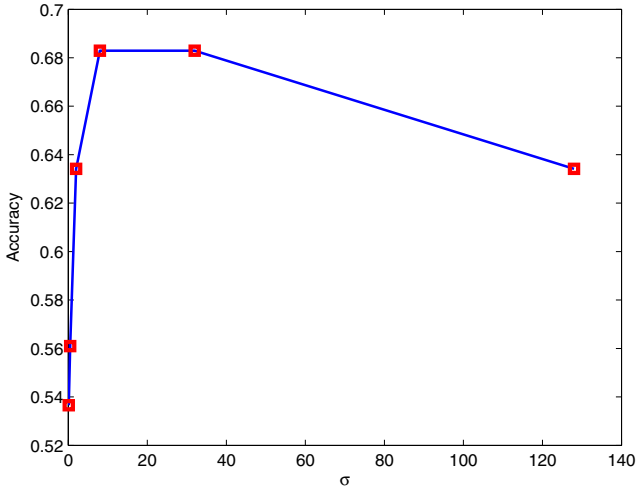

Fig. 9 The effect of parameter σ on the performance

Similarly, we can obtain that in multi-class classification case, $\chi_F^2 = 16.1700$ and $F_F = 10.5228$. It is also clear that the value of F_F is much larger than critical value. From Table 5, we can see that the average ranking of MRMMTSSVM is the lowest. They both represent that MRMMTSSVM has better experimental performance than other three algorithms.

In conclusion, our approaches are more valid than other algorithms both in binary and multi-class classification condition.

5.3 The effect of parameters on the performance

In Ramp-MMTSSVM and MRMMTSSVM, there are three parameters, i.e., k, ν, and σ. The different values of the three parameters will have a great influence on experimental accuracies. Therefore, in this part, we conduct experiments on Sonar dataset to investigate the influence of the three parameters on experimental performance. The results are shown in Figs. 7, 8 and 9.

The x-axis of the three figures denotes the changing values of the three parameters, and the y-axis denotes the corresponding accuracies. From Fig. 7, we can see that on Sonar dataset when k is from 0.5 to 1.5, the accuracy increases. While when $k = 2.0$, the accuracy decreases. Figure 8 shows that when ν is small, the accuracy is stable and great change of accuracy takes place when ν increases. We can get in Fig. 9 that the accuracy changes greatly when σ is small and as σ increases, the accuracy tends to be stable. In conclusion, the three parameters k, ν and σ all have an influence on experimental results in Ramp-MMTSSVM and MRMMTSSVM. In addition, the proper ranges of these paremeters are different for different datasets.

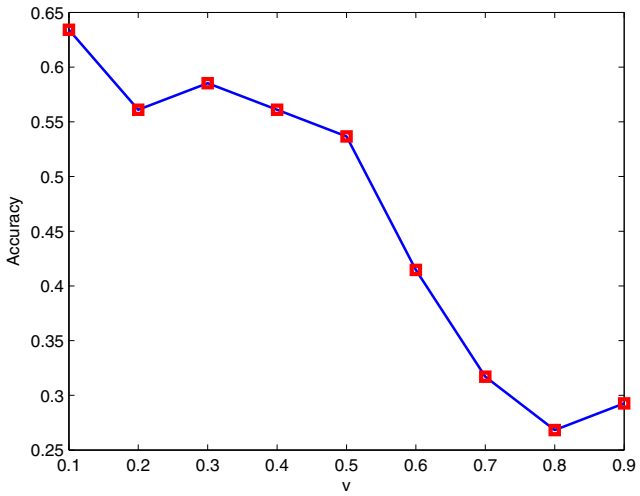

Fig. 8 The effect of parameter ν on the performance

6 Conclusions

In this paper, we propose a Ramp loss maximum margin of twin spheres support vector machine. It reduces the sensitivity to outliers and improves generalization performance. Because it is a non-differentiable non-convex optimization problem, we adopt CCCP approach to solve it. Furthermore, we prove the properties of the parameters ν_1 and ν_2, and testify them by one artificial experiment. In addition, we extend Ramp-MMTSSVM to multi-class classification problems by RVO strategy. The experimental results show that our approaches both in binary and multi-class classification cases have better performance than other algorithms on twenty benchmark datasets. During the experiments, we find that the computational speed of

our algorithms is a little slower than other algorithms. Therefore, how to accelerate the computational speed is our future work.

Acknowledgments This work was supported in part by National Natural Science Foundation of China (No.11671010).

References

1. Ripley BD (1996) Pattern recognition and neural networks. Cambridge University Press, Cambridge
2. Meza J, Espitia H, Montenegro C, Crespo RG (2016) Statistical analysis of a multi-objective optimization algorithm based on a model of particles with vorticity behavior. Soft Comput 20(9):3521–3536
3. Vapnik V (1995) The nature of statistical learning theory. Springer, New York
4. Zhang W, Yoshida T, Tang X (2008) Text classification based on multi-word with support vector machine. Knowl-Based Syst 21(8):879–886
5. Ghosh S, Mondal S, Ghosh B (2014) A comparative study of breast cancer detection based on SVM and MLP BPN classifier. In: First international conference on automation, control, energy & systems (ACES-14), pp 87–90
6. Gohariyan E, Esmaeilpour M, Shirmohammadi MM (2017) The combination of mammography and MRI for diagnosing breast cancer using fuzzy NN and SVM. Int J Interact Multimed Artif Intell 4(5):20–24
7. Naz S, Ziauddin S, Shahid AR (2018) Driver fatigue detection using mean intensity, SVM, and SIFT. Int J Interact Multimed Artif Intell 5(IP):1
8. Pang Y, Zhang K, Yuan Y, Wang K (2014) Distributed object detection with linear SVMs. IEEE T Cybern 44(11):2122–2133
9. Jayadeva KhemchandaniR, Chandra S (2007) Twin support vector machines for pattern classification. IEEE T Pattern Anal 29(5):905–910
10. Peng X (2010) A ν-twin support vector machine (ν-TSVM) classifier and its geometric algorithms. Inform Sciences 180(20):3863–3875
11. Xie X, Sun S (2014) Multi-view Laplacian twin support vector machines. Appl Intell 41(4):1059–1068
12. Xu Y, Guo R (2014) An improved ν-twin support vector machine. Appl Intell 41(1):42–54
13. Wang H, Zhou Z (2017) An improved rough margin-based ν-twin bounded support vector machine. Knowl-Based Syst 128:125–138
14. Wang H, Zhou Z, Xu Y (2018) An improved ν-twin bounded support vector machine. Appl Intell 48(4):1041–1053
15. Xu Y, Wang L, Zhong P (2012) A rough margin-based ν-twin support vector machine. Neural Comput Appl 21:1307–1317
16. Xu Y, Yu J, Zhang Y (2014) KNN-Based weighted rough ν-twin support vector machine. Knowel-Based Syst 71:303–313
17. Kumar MA, Gopal M (2009) Least squares twin support vector machines for pattern classification. Expert Syst Appl 36(4):7535–7543
18. Peng X, Xu D (2013) A twin-hypersphere support vector machine classifier and the fast learning algorithm. Inform Sci 221:12–27
19. Xu Y (2016) Maximum margin of twin spheres support vector machine for imbalanced data classification. IEEE T Cybern 47(6):1540–1550
20. Huang X, Shi L, Suykens JAK (2014) Ramp loss linear programming support vector machine. J Mach Learn Res 15:2185–2211
21. Yuille A, Rangarajan A (2003) The concave-convex procedure. Neural Comput 15:915–936
22. Liu D, Shi Y, Huang X (2016) Ramp loss least squares support vector machine. J Comput Sci-Neth 14:61–68
23. Zhang Z, Krawczyk B, Garcìa S, Rosales- Pérez A, Herrera F (2016) Empowering one-vs-one decomposition with ensemble learning for multi-class imbalanced data. Knowl-Based Syst 106:251–263
24. Zhou L, Wang Q, Fujita H (2017) One versus one multi-class classification fusion using optimizing decision directed acyclic graph for predicting listing status of companies. Inform Fusion 36:80–89
25. Yang X, Yu Q, Guo T (2013) The one-against-all partition based binary tree support vector machine algorithms for multi-class classification. Neurocomputing 113:1–7
26. Tomar D, Agarwal S (2015) A comparison on multi-class classification methods based on least squares twin support vector machine. Knowl-Based Syst 81:131–147
27. Xu Y, Guo R (2014) A twin hyper-sphere multi-class classification support vector machine. J Intell Fuzzy Syst 27(4):1783–1790
28. Kumar D, Thakur M (2018) All-in-one multicategory least squares nonparallel hyperplanes support vector machine. Pattern Recogn Lett 105:165–174
29. Angulo C, Parra X, Català A (2003) K-SVCR. a support vector machine for multi-class classification. Neurocomputing 55(1-2):57–77
30. Xu Y (2016) K-nearest neighbor-based weighted multi-class twin support vector machine. Neurocomputing 205:430–438
31. Xu Y, Guo R, Wang L (2013) A twin multi-class classification support vector machine. Cogn Comput 5(4):580–588

Sijie Lu She is currently pursuing the bachelor's degree in College of Science, China Agriculture University. Her research interests include support vector machine and data mining.

Huiru Wang received her B.S. and M.S. degrees in College of Science from China Agricultural University in 2013 and 2015, respectively. She is currently pursuing the Ph.D. degree in College of Science, China Agriculture University. Her research interests include support vector machine and data mining.

Zhijian Zhou received her Ph.D. degree in College of Engineering from China Agricultural University in 2007. Dr. Zhou is now a Professor of College of Science, China Agricultural University. Her research interests include machine learning and data mining.