International Journal of Machine Learning and Cybernetics (2022) 13:3341–3358
https://doi.org/10.1007/s13042-022-01598-4

ORIGINAL ARTICLE

Risk-Averse support vector classifier machine via moments penalization

Cui Fu[1] · Shuisheng Zhou[1] · Junna Zhang[1] · Banghe Han[1] · Yuxue Chen[1] · Feng Ye[1]

Received: 14 September 2021 / Accepted: 10 June 2022 / Published online: 7 July 2022

Abstract
Support vector machine (SVM) has always been one of the most successful learning methods, with the idea of structural risk minimization which minimizes the upper bound of the generalization error. Recently, a tighter upper bound of the generalization error, related to the variance of loss, is proved as the empirical Bernstein bound. Based on this result, we propose a novel risk-averse support vector classifier machine (RA-SVCM), which can achieve a better generalization performance by considering the second order statistical information of loss function. It minimizes the empirical first- and second-moments of loss function, i.e., the mean and variance of loss function, to achieve the "right" bias-variance trade-off for general classes. The proposed method can be solved by the kernel reduced and Newton-type technique under certain conditions. Empirical studies show that the RA-SVCM achieves the best performance in comparison with other classical and state of art methods. The additional analysis shows that the proposed method is insensitive to the parameters, so abroad range of parameters lead to satisfactory performance. The proposed method is a general form of standard SVM, so it enriches the related studies of SVM.

Keywords Bennetts's inequality · Empirical Bernstein bounds · SVM · Risk-Averse SVCM · Representation theorem · Moment penalization · Newton algorithms

1 Introduction

Support Vector Machine (SVM) [1, 2] and its extensions have always been one of the most successful machine leaning methods for supervised learning due to their accuracy, robustness and indifference towards the instance data type. They are widely used in classification and regression problems, such as face recognition [3, 4], spam recognition [5], handwriting number recognition [6], disease diagnosis [7–9], and pattern recognition [10], etc.

During the past few decades, many common improved methods based on SVM have been developed well, among which the most methods change the regularization terms, constraints, and loss functions to improve the learning ability of the model. For instance, the least-square SVM (LS-SVM) [11] method can be easily implemented due to the utilization of the equality constraints. A radius-margin-based SVM model with LogDet regularization considers the radius and introduce a negative LogDet term to improve the model accuracy [12]. A risk-averse classifier allows for associating distinct risk functional to each classes [13]. The sparse LSSVM in primal using cholesky factorization for large-scale problems and the random reduced P-LSSVM (RRP-LSSVM) achieves the sparse solutions of LSSVM [14]. The twin Support vectors [15–17] finds two hyperplanes, one for each class, and classifies points according to which hyperplane a given point is closest to. These methods are mainly based on margin theory [1] or structural risk minimization theory. They focus on margin of a few instances or first-order information of loss function instead of the characteristics of the data itself.

There are a few studies considered the effect of the information about data on the generalization ability of SVM-style algorithms. The SVM+ approach can lower the overall system's VC-dimension and hence attain better generalization by taking advantage of the structure in the training data [18, 19]. But this model still consider only the samples that are in the class boundaries regardless of class distribution characteristics. The Support Vector Machines with multiview Privileged improve the performance of the classification tasks by exploiting the complementary information among multiple

✉ Shuisheng Zhou
sszhou@mail.xidian.edu.cn

[1] School of Mathematics and Statistics, Xi'dian University, Xi'an 710071, China

feature sets [20–22]. The margin distribution optimization (MDO) algorithm [23] optimizes margin distribution by minimizing the sum of exponential loss, but this method tends to get a local minima with slow convergence since the objective function is non-differential and non-differential. At the same time, more and more attention has been paid to the second-order statistical characteristics of training datas. A robust least-squares SVM [24] with minimization of mean and variance of modeling error distributes smaller weight to larger error training samples and lager weight to small error training samples, which is more robust in regards to random noise. The large margin distribution machine (LDM) [25] tries to optimize the margin distribution by maximizing the margin mean and minimizing the margin variance simultaneously. The optimal margin distribution machine (ODM) [26], which is a simpler but powerful formulation than LDM, were successively proposed by simplifing the variance term of LDM and introducing a insensitive loss. All studies of SVM above, however, focused on margin-based explanation or partial information of training data, whereas the influence of the loss distribution for SVM has not been well exploited.

Based on Hoeffding's inequality, the SVM minimizes the structural risk that is upper bound of the generalization error. The empirical Bernstein bounds [27], however, disclosed that minimize the structural risk does not necessarily lead to better generalization performance, and instead the variance of loss function has been proven be more crucial. In other words, the empirical Bernstein bounds is a tighter upper bound of the generalization error. Inspired by the theory above, we propose a Risk-Averse support vector classifier machine(RA-SVCM). The RA-SVCM tries to achieve the "right" bias-variance trade-off for general classes by minimizing the empirical first- and second-moments of loss simultaneously, i.e., the mean and variance of loss. Comprehensive experiments on twelve regular scale data sets and eight large scale data sets show the superiority of RA-SVCM to SVM and many state-of-the-art methods, verifying that the minimum variance of loss function is more crucial for SVM-style learning approaches than minimum structural risk.

The remainder of this paper is organized as follows. An overview of the related work is introduced in Sect. 2. In Sect. 3, the details of the developed RA-SVCM are stated. In Sect. 4, the differences with related methods are presented. the details of the developed RA-SVCM are stated. The experimental results are then reported in Sect. 5 to validate the validity of the method. Finally, Sect. 6 concludes this paper.

2 Related work

2.1 SVM

This section briefly introduces SVM. For convenience, we first introduce some notations which are used throughout the paper. Let $\mathcal{X} \subset \mathbb{R}^d$ and $\mathcal{Y} = \{+1, -1\}$ denote the input space and output space, respectively. Denote by $\mathcal{D}$ an (unknown) underlying probability distribution over the product space $\mathcal{X} \times \mathcal{Y}$. Let $u_+ = \max\{0, u\}$. For training set $S = \{(\boldsymbol{x}_1, y_1) \cdots (\boldsymbol{x}_m, y_m)\} \in \{\mathcal{X} \times \mathcal{Y}\}^m$, which drawn independently and identically ($i.i.d$) according to the distribution $\mathcal{D}$, the goal of soft SVM is to learn parameters $(\boldsymbol{w}, b)$ of a hypothesis $f(\boldsymbol{x}) = \langle \boldsymbol{w}, \phi(\boldsymbol{x}) \rangle + b$ from the optimization problem:

$$\min_{\boldsymbol{w},b} \frac{C}{m} \sum_{i=1}^{m} [1 - y_i(\langle \boldsymbol{w}, \phi(\boldsymbol{x}_i) \rangle + b)]_+ + \frac{1}{2}\|\boldsymbol{w}\|^2, \tag{1}$$

where $\boldsymbol{w} \in \mathbb{H}, b \in \mathbb{R}$, tradeoff paramete $C > 0$ and $\phi(\cdot)$ maps $\boldsymbol{x}_i$ to a high dimensional feature space. $\mathbb{H}$ is the reproducing kernel Hilbert space (RKHS) associated with a kernel fuction $k : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ satisfying $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left\langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \right\rangle$. For convenience, SVM for learning a homgenous halfpace is considered, where the bias term b is set to be zero. Because Steinwart et al. [28] proved that SVMs without offset term b have convergence rates and classification performance that are comparable to SVMs with offset, while the absence of the offset gives more freedom in the algorithm design.

On the basis of the representation theorem [29–31], the learning result $\boldsymbol{w}$ can be represented by a linear combination of the kernel functions:

$$\boldsymbol{w} = \sum_{i=1}^{m} \boldsymbol{\alpha}_i \phi(\boldsymbol{x}_i), \tag{2}$$

which is a finite linear combination of $\phi(\boldsymbol{x}_i)$. Therefore, we can optimize problem (1) with respect to the coefficients $\boldsymbol{\alpha}$ in $\mathbb{R}^m$ instead of the parameters $\boldsymbol{w}$ in $\mathbb{H}$ as follows.

$$\min_{\alpha \in \mathbb{R}^m} \frac{C}{m} \sum_{i=1}^{m} \left(1 - y_i K_i \boldsymbol{\alpha}\right)_+ + \frac{1}{2} \boldsymbol{\alpha}^\top K \boldsymbol{\alpha}, \tag{3}$$

where the kernel matrix K satisfies $K_{i,j} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ $\left\langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \right\rangle$ and K_i is the i-th row of K. It is infeasible to get the whole kernel matrix when the sample size m is larger enough. According to sparse representation theorem [32, 33], the learning result can be represented as

$$\boldsymbol{w} = \sum_{i \in J} \boldsymbol{\alpha}_i \phi(\boldsymbol{x}_i), \tag{4}$$

where a reduced set J is selected randomly from the index set $M = \{1, 2, ...m\}$, and $|J| \leq 0.1m$ [33, 34]. So the reduced version of standard SVM corresponding to (3) can be represented as

$$\min_{\boldsymbol{\alpha}_J \in \mathbb{R}^{|J|}} \frac{C}{m} \sum_{i=1}^{m} \left(1 - y_i K_{Ji} \boldsymbol{\alpha}_J\right)_+ + \frac{1}{2} \boldsymbol{\alpha}_J^\top K_{JJ} \boldsymbol{\alpha}_J. \tag{5}$$

Here, K_{JJ} is the sub-matrix of K, whose elements are $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ for $i \in J$ and $j \in J$. K_{JM} is a sub-matrix of K, whose elements are $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ for $i \in J$ and $j \in M$ and K_{Ji} is i-th column of K_{JM}.

2.2 LDM and ODM

Based on theoretical results that the margin distribution is more crucial to the generalization performance, the large margin distribution Machine (LDM) [25] is proposed to achieving a better generalization performance by optimizing the margin distribution of model characterized by the first and second order statistics, i.e., the margin mean and variance. The formulation of LDM is as following.

$$\begin{aligned} &\min_{\boldsymbol{w}, \xi_i} \tfrac{1}{2} \|\boldsymbol{w}\|^2 + \mu_1 \gamma_v - \mu_2 \gamma_m + \tfrac{\lambda}{m} \sum_{i=1}^{m} \xi_i \\ &s.t. y_i \boldsymbol{w}^T \phi(\boldsymbol{x}_i) \geq 1 - \xi_i, \xi_i \geq 0, \forall i, \end{aligned} \tag{6}$$

where $\gamma_m = \frac{1}{m} \sum_{i=1}^{m} y_i \boldsymbol{w}^T \phi(\boldsymbol{x}_i)$ and $\gamma_v = \frac{1}{m} \sum_{i=1}^{m} (y_i \boldsymbol{w}^T \phi(\boldsymbol{x}_i) - \gamma_m)^2$ are the margin mean and margin variance respectively.

The optimal margin distribution machine (ODM) [26], which is a simpler but powerful formulation than LDM, were successively proposed by simplifying the variance term of LDM and introducing a insensitive loss.

$$\begin{aligned} &\min_{\boldsymbol{w}, \xi_i, \varepsilon_i} \tfrac{1}{2} \|\boldsymbol{w}\|^2 + \tfrac{\lambda}{m} \sum_{i=1}^{m} \tfrac{\xi_i^2 + \mu \varepsilon_i^2}{(1-D)^2}, \\ &s.t. \quad y_i \boldsymbol{w}^T \phi(\boldsymbol{x}_i) \geq 1 - D - \xi_i, \\ &\quad y_i \boldsymbol{w}^T \phi(\boldsymbol{x}_i) \leq 1 + D + \varepsilon_i, \forall i. \end{aligned} \tag{7}$$

where λ and μ are trading-off parameters, and D is a parameter for controlling the number of support vectors. For kernel ODM, the objective function of primal problem (7) can be reformulated as the following form,

$$\begin{aligned} f_O(w) = &\frac{1}{2} \|\boldsymbol{w}\|^2 + \frac{\lambda}{m(1-D)^2} (\sum_{i=1}^{m} (1 - D - y_i \boldsymbol{w}^T \phi(\boldsymbol{x}_i))_+^2 \\ &+ \mu \sum_{i=1}^{m} (y_i \boldsymbol{w}^T \phi(\boldsymbol{x}_i) - 1 - D)_+^2). \end{aligned} \tag{8}$$

3 The proposed Risk-Averse SVCM

In this section, we begin with a discussion of the confidence bounds most frequently used in learning theory.

Suppose this underlying observation is modeled by a random variable $(\boldsymbol{x}, y)$ distributed in some space $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ according to law $\mathcal{D}$, then this underlying observation can be denoted as $l(y, f(\boldsymbol{x}))$. The $\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[l(y, f(\boldsymbol{x}))]$ and $\frac{1}{m} \sum_{i=1}^{m} l(y_i, f(\boldsymbol{x}_i))$ are called expected risk and empirical risk of hypothesis f respectively.

According to the above-mentioned conditions and the related theorems in [27], some results can be given as:

Suppose that $\mathcal{D}$ is a distribution over $\mathcal{X} \times \mathcal{Y}$ such that with probability 1 we have that $\|\boldsymbol{x}\|_2 \leq R$. Let $l : \mathcal{H} \times \mathcal{Z} \to \mathbb{R}$ be a loss function, and $l(y_i, f(\boldsymbol{x}_i))$ be a sequence of *i.i.d.* random variables with values in [0, 1]. Then for any $\delta > 0$, with probability at least $1 - \delta$ we have

$$\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\, l(y, f(\boldsymbol{x})) \leq \frac{1}{m} \sum_{i=1}^{m} l(y_i, f(\boldsymbol{x}_i)) + \sqrt{\frac{\ln 1/\delta}{2m}}, \tag{9}$$

The inequation of (9) cited Hoeffding's inequality probability in form of a confidence dependent bound on the deviation [35]. A drawback of this inequality is that the confidence interval is independent of the hypothesis in question, and of order $\sqrt{1/m}$.

And for any $\delta > 0$, then with probability at least $1 - \delta$ we aslo have

$$\begin{aligned} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\, l(y, f(\boldsymbol{x})) \leq &\frac{1}{m} \sum_{i=1}^{m} l(y_i, f(\boldsymbol{x}_i)) + \sqrt{\frac{2\mathbb{V}(l) \ln 1/\delta}{m}} \\ &+ \frac{\ln 1/\delta}{3m}, \end{aligned} \tag{10}$$

where $\mathbb{V}(l) = \mathbb{E}(l - El)^2$. The inequality of (10) is called Bennetts's inequality, and the confidence interval of this inequality becomes $2\sqrt{\mathbb{V}l}$ times the confidence interval of the Hoeffding's inequality. This bound proves us of higher accuracy for hypotheses of small variance, and of lower accuracy for hypotheses of large variance. But the first term on the right hand side depends the unmeasurable variance, leaving us with a uniformly blurred view of the hypothesis class. So Maurer and Pontil [27] provide empirical Bernstein bounds (11), which is a purely data-dependent bound with similar properties as Bennetts's inequality. This bound makes the diameter of the confidence interval observable and provides us with a view of the loss class which is more in focus for hypotheses of small sample variance.

$$\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\, l(y, f(\boldsymbol{x})) \leq \frac{1}{m} \sum_{i=1}^{m} l(y_i, f(\boldsymbol{x}_i)) + \sqrt{\frac{2\mathbb{V}_m(l) \ln 2/\delta}{m}} + \frac{7 \ln 2/\delta}{3(m-1)} \tag{11}$$

where $\mathbb{V}(l) = \mathbb{E}(l - El)^2$, $\mathbb{V}_m(l) = \frac{1}{m-1} \sum_{i=1}^{m} (l_i - \bar{l})^2$, and $l_i = l(y_i, f(\boldsymbol{x}_i))$. Minimizing this uniform convergence bound leads to the sample variance penalization principle:

$$\arg\min_{f\in\mathcal{F}} \frac{1}{m}\sum_{i=1}^{m} l(y_i,f(\boldsymbol{x}_i)) + \tau\sqrt{\frac{V[l(y,f(\boldsymbol{x}))]}{m}} \tag{12}$$

Based on this principle,we propose a new model, which allows for a reduction of mean of the hinge loss as well as the minimization of its variance. As the variance of model loss characterize the stability of the model at sample datas, we named it Risk-Averse support vector classifier machine(RA-SVCM).

Here we first analyze the relevant loss function. Given an *i.i.d.* training set $\{\boldsymbol{x}_i, y_i\}_{i=1}^{m}$, our aim is to minimize the future probability of error $\mathbb{P}_{\mathcal{D}}[yf(\boldsymbol{x}) \le 0]$ in classification problems. Note that $\mathbb{E}[\mathbf{I}_{yf(\boldsymbol{x})\le 0}] = \mathbb{P}_{\mathcal{D}}[yf(\boldsymbol{x}) \le 0]$, we can minimize the future probability of error by minimizing the mean of 0-1 loss:

$$l(y,f(\boldsymbol{x})) = I_{yf(\boldsymbol{x})\le 0} = \begin{cases} 0, & yf(\boldsymbol{x}) > 0 \\ 1, & yf(\boldsymbol{x}) \le 0. \end{cases} \tag{13}$$

However, the empirical risk minimization with the 0-1 loss is a difficult problem to deal with. Towards this end, we replace the 0-1 loss by the hinge loss :

$$l(y,f(\boldsymbol{x})) = \max\{0, 1 - yf(\boldsymbol{x})\} = \begin{cases} 0 & yf(\boldsymbol{x}) > 1 \\ 1 - yf(\boldsymbol{x}) & yh(\boldsymbol{x}) \le 1, \end{cases} \tag{14}$$

which is a convex approximation of the 0–1 loss. Therefore, we relate the future probability of error to the hinge loss:

$$\mathbb{P}_{\mathcal{D}}[yf(\boldsymbol{x}) \le 0] = \mathbb{E}[\mathbf{I}_{yf(\boldsymbol{x})\le 0}] \le E_{\mathcal{D}}[(1 - yf(\boldsymbol{x}))_+]. \tag{15}$$

That is, we can minimize the mean and standard deviation of hinge loss to improve the performance of the SVMs as follows:

$$\min_{f\in\mathcal{F}} \frac{1}{m}\sum_{i=1}^{m}(1 - y_i f(\boldsymbol{x}_i))_+ + \tau\sqrt{\frac{\sum_{i=1}^{m}\left((1 - y_i f(\boldsymbol{x}_i))_+ - \frac{1}{m}\sum_{j=1}^{m}(1 - y_j f(\boldsymbol{x}_j))_+\right)^2}{m(m-1)}}. \tag{16}$$

Since there is an unknown trade-off parameter between the two terms in (16), and it is difficult to solve because of the standard deviation, we can minimize that cost by the following problem:

$$\begin{aligned} &\min_{f\in\mathcal{F}} \frac{1}{m}\sum_{i=1}^{m}(1 - y_i f(\boldsymbol{x}_i))_+ \\ &s.t. \sqrt{\frac{\sum_{i=1}^{m}\left((1-y_i f(\boldsymbol{x}_i))_+ - \frac{1}{m}\sum_{j=1}^{m}(1-y_j f(\boldsymbol{x}_j))_+\right)^2}{m(m-1)}} \le B, \end{aligned} \tag{17}$$

where $\mathcal{F}$ is a finite class of hypotheses $f : \mathcal{X} \to [0, 1]$, and the trade-off parameter is now parametrized by B. For every τ, there is a B that obtains the same optimal function. In particular, $B = \infty$ is equivalent to $\tau = 0$. The problem above can be equivalent to the problem as follows:

$$\begin{aligned} &\min_{f\in\mathcal{F}} \frac{\mu_1}{m}\sum_{i=1}^{m}(1 - y_i f(\boldsymbol{x}_i))_+ \\ &s.t. \frac{\sum_{i=1}^{m}\left((1-y_i f(\boldsymbol{x}_i))_+ - \frac{1}{m}\sum_{j=1}^{m}(1-y_j f(\boldsymbol{x}_j))_+\right)^2}{m(m-1)} \le B^2, \end{aligned} \tag{18}$$

where $\mu_1 \ge 0$ is trade-off parameters of mean term. By introducing the Lagrange multipliers μ_2 for the constraints, the Lagrange of Eq.(19) lead to

$$\frac{\mu_1}{m}\sum_{i=1}^{m}(1-y_i f(\boldsymbol{x}_i))_+ + \frac{\mu_2}{m(m-1)}\sum_{i=1}^{m}\left((1 - y_i f(\boldsymbol{x}_i))_+ - \frac{1}{m}\sum_{j=1}^{m}(1 - y_j f(\boldsymbol{x}_j))_+\right)^2 - \mu_2 B^2. \tag{19}$$

Since the trird term does not involve the function f, we can merely optimize the first two items, and consider model regularization term, we can get new model as follows:

$$\min_{f\in\mathcal{F}} \lambda_1\sum_{i=1}^{m}(1 - y_i f(\boldsymbol{x}_i))_+ + \lambda_2\sum_{i=1}^{m}\left((1 - y_i f(\boldsymbol{x}_i))_+ - \frac{1}{m}\sum_{j=1}^{m}(1 - y_j f(\boldsymbol{x}_j))_+\right)^2 + R(f) \tag{20}$$

where $\lambda_1 = \frac{\mu_1}{m}$, $\lambda_2 = \frac{\mu_2}{m(m-1)}$. This is the robust model we propose, and the specific forms of which are described in the next section.

3.1 New objective function of RA-SVCM

The objective function with the minimized mean and variance of the loss is constructed in order to improve generalization performance of SVM.

The new objective function of nonlinear RA-SVCM is

$$\min_{\alpha\in\mathbb{R}^m} \lambda_1 \sum_{i=1}^{m} (\boldsymbol{r}_i)_+ + \lambda_2 \sum_{i=1}^{m} \left((\boldsymbol{r}_i)_+ - \frac{1}{m} \sum_{j=1}^{m} (\boldsymbol{r}_j)_+ \right)^2 + \frac{1}{2}\boldsymbol{\alpha}^\top K \boldsymbol{\alpha}, \tag{21}$$

where $\boldsymbol{r}_i = 1 - y_i K_i \boldsymbol{\alpha}$. λ_1 and λ_2 are the regularization parameters, which are determined by using the cross-validation method [36, 37].

The new objective function has the following features:

1) The first term $\lambda_1 \sum_{i=1}^{m} (\boldsymbol{r}_i)_+$ is empirical risk that is minimized to make the points generated by the model closer to the sample datas, hence it can minimize the entirety of classification error in the training phase.
2) The second term $\lambda_2 \sum_{i=1}^{m} \left((\boldsymbol{r}_i)_+ - \frac{1}{m} \sum_{j=1}^{m} (\boldsymbol{r}_j)_+ \right)^2$ is the variance regularization that acts as a global loss stabilizer factor. This global loss adjusting factor is used to coordinate all of the samples in order to achieve a better modeling performance. It is clear that minimizing this term can improve the generalization performance of modeling based on the empirical Bernstein bounds. Obviously, when λ_2 is equal to zero, the objective function of (21) is same as that of standard SVM. This means that the standard SVM is a special case of RA-SVCM.
3) The third term $\frac{1}{2}\boldsymbol{\alpha}^\top K \boldsymbol{\alpha}$ is the model regularization term that is used to avoid overfitting of the model.

It is evident that the regularization term in this new objective function includes both the variance and model regularization term as show in Fig. 1. It is also well-known that there is an increase in the classification accuracy when the mean of the loss is minimized, and that minimizing its variance can lead to a classifier with higher generalization ability.

For approximately separable dataset, the average loss of SVM tends to zero. Considering that the second-order central moment of loss is similar to the second-order origin moment, the linear and nonlinear RA-SVCM can be simplified as follow:

$$\min_{\alpha\in\mathbb{R}^m} \lambda_1 \sum_{i=1}^{m} (\boldsymbol{r}_i)_+ + \lambda_2 \sum_{i=1}^{m} (\boldsymbol{r}_i)_+^2 + \frac{1}{2}\boldsymbol{\alpha}^T K \boldsymbol{\alpha}. \tag{22}$$

These models are denoted as sRA-SVCM. When $\lambda_2 \to 0$, the objective function of (22) is same as it is when using the standard SVM with hinge loss (SVM-H). And when $\lambda_1 \to 0$, the objective function of (22) is same as it is when using the standard SVM with squared hinge loss (SVM-SH),

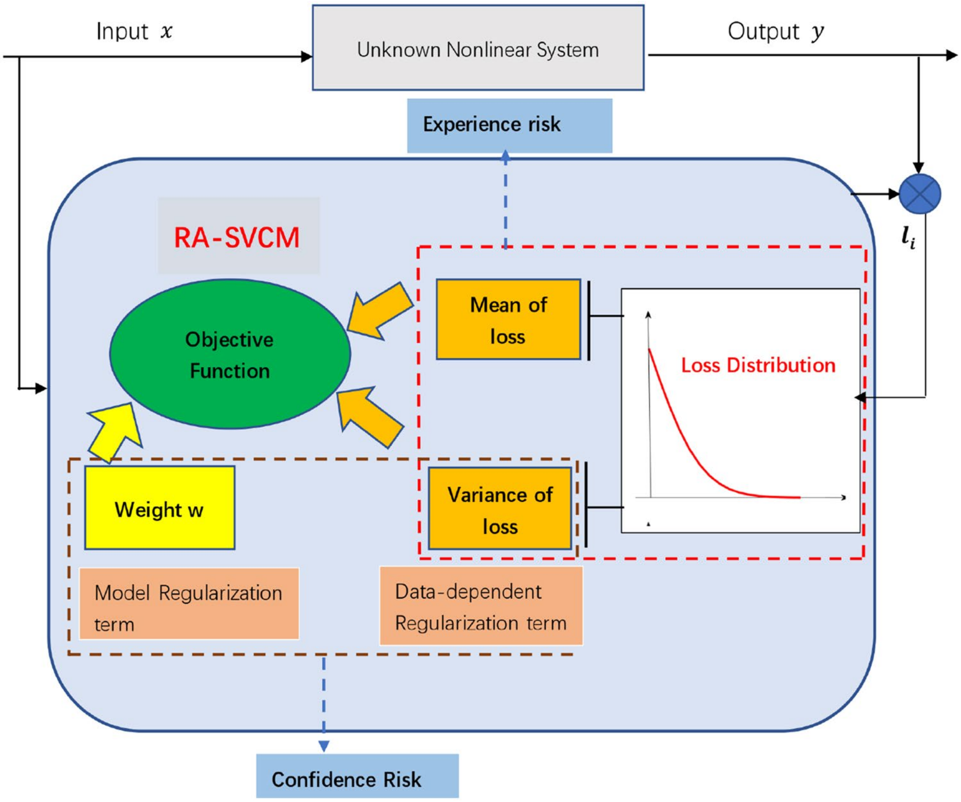


Fig. 1 RA-SVCM

this means that both the standard SVM with hinge loss and squared hinge loss are special cases of sRA-SVCM.

3.2 Theoretical analysis

In this section, we study the statistical property of RA-SVCM. By applying the result from empirical bernstein bounds (11), for our model, i.e., $f(x) = \langle \boldsymbol{w}, \psi(\boldsymbol{x}) \rangle = G_i^T \boldsymbol{\alpha}$, we can get a result as follows:

Theorem 1 *Suppose that* $(\boldsymbol{x}_i, y_i)_{i=1}^m$ *be drawn i.i.d. according to the distribution* $\mathcal{D}$ *such that with probability* 1 *we have that* $\|\boldsymbol{x}\|_2 \leq R$, *and let* $\|\boldsymbol{\alpha}\|_2 \leq C$, *and the Gaussian kernel function* $G_{i,j} = K(x_i, x_j) = \exp(-h\left\|x_i - x_j\right\|^2)$ *is bounded in the feature spaces, i.e.,* $\exists M > 0$ *and* $c > 0$ *such that* $\max_{G_i^T \boldsymbol{\alpha} \in [-M,M]}\{0, 1 - yG_i^T \boldsymbol{\alpha}\} \leq c$. *Then, for any* $\delta > 0$, *with probability at least* $1 - \delta, \forall \boldsymbol{\alpha} \in R^m$,

$$\mathbb{E}_{\mathcal{D}}[(1 - yG_i^T \boldsymbol{\alpha})_+] \leq \hat{p} + \sqrt{\frac{18V[(1 - yG_i^T \boldsymbol{\alpha})_+] \ln(\mathcal{M}(m)/\delta)}{m}} + \frac{15c \ln(\mathcal{M}(m)/\delta)}{m - 1}, \tag{23}$$

where $\hat{p} = \hat{P}[l(y, G_i^T \boldsymbol{\alpha})] = \frac{1}{m} \sum_{i=1}^m (1 - yG_i^T \boldsymbol{\alpha})_+$ *is the empirical loss, and an extra c appears in the third term to normalize the hinge loss so that it has the range* [0, 1].

Proof Since $\|\boldsymbol{\alpha}\|_2 \leq C$, $\|\boldsymbol{x}\|_2 \leq R$, then the Gaussian kernel is bounded on $\mathcal{X}$, thus $\exists M > 0$, such that $G_i^T \boldsymbol{\alpha} \in [-M, M]$. So $\exists c > 0$, such that the loss function $l_1(G_i^T \boldsymbol{\alpha}, y) = (1 - yG_i^T \boldsymbol{\alpha})_+ \in [0, c]$. Then we can apply the empirical bernstein inequation (11) on hinge loss which is normalized. □

It is clear that the bounds of theorem 1 has estimation errors which can be as small as $O(1/n)$ for small sample variances, while the bound of Hoeffdings inequality on which SVMs are based is of order $1/\sqrt{n}$.

3.3 Solutions to RA-SVCM and sRA-SVCM

In this section, Newton-type algorithms, which has quadratic convergence rate, is used to train the RA-SVCM and sRA-SVCM. However, due to the hinge loss is not differentiable, the Newton-type methods do not work for it directly. Thus, some smooth loss function are chosen to approximate the hinge loss, such as least squares loss [11, 38], logistic loss [33] and Huber loss [1, 34]. According to SSVM presented by Lee and Mangasarian [32] and a Smoothing SVM with efficient reduced techniques proposed by Zhou [39], the logistic loss $\varphi_p(r) = \frac{1}{p} \log(1 + \exp(pr))$ is adopted to approximate hinge loss in this paper. To overcome any potential overflowing, a stable form is given as $\varphi_p(r) = \max\{r, 0\} + \frac{1}{p} \log(1 + \exp(-|pr|))$.

For convenience, here we only introduce the solution of nonlinear models. The smooth forms of problem (21) and (22) can be formulated as:

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^m} f_1(\boldsymbol{\alpha}) := \lambda_1 \sum_{i=1}^m \varphi_p(\boldsymbol{r}_i) + \lambda_2 \sum_{i=1}^m (\varphi_p(\boldsymbol{r}_i) - \frac{1}{m} \sum_{j=1}^m \varphi_p(\boldsymbol{r}_j))^2 + \frac{1}{2} \boldsymbol{\alpha}^\top K \boldsymbol{\alpha}, \tag{24}$$

and

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^m} f_2(\boldsymbol{\alpha}) := \lambda_1 \sum_{i=1}^m \varphi_p(\boldsymbol{r}_i) + \lambda_2 \sum_{i=1}^m (\varphi_p(\boldsymbol{r}_i))^2 + \frac{1}{2} \boldsymbol{\alpha}^\top K \boldsymbol{\alpha}, \tag{25}$$

respectively.

The gradient $\nabla f_1(\boldsymbol{\alpha}^t)$ and Hessian matrix $\nabla^2 f_1(\boldsymbol{\alpha}^t)$ of the objective function in problem (24) are

$$\nabla f_1(\boldsymbol{\alpha}) = K_{I_1} \boldsymbol{u_1} + K_{I_2} \boldsymbol{u_2} + K \boldsymbol{\alpha}, \tag{26}$$

and

$$\nabla^2 f_1(\boldsymbol{\alpha}) = K_{I_3}{}^\top \Lambda_{I_3} K_{I_3} + 2\lambda_2 K_{I_1}{}^\top K_{I_1} - \frac{2\lambda_2}{m} \boldsymbol{q}\boldsymbol{q}^\top + K. \tag{27}$$

The gradient $\nabla f_2(\boldsymbol{\alpha}^t)$ and Hessian matrix $\nabla^2 f_2(\boldsymbol{\alpha}^t)$ of the objective function in problem (25) are

$$\nabla f_2(\boldsymbol{\alpha}) = K_{I_1} \boldsymbol{u_1} + K_{I_3} \boldsymbol{u_3} + K \boldsymbol{\alpha}, \tag{28}$$

and

$$\nabla^2 f_2(\boldsymbol{\alpha}) = K_{I_3}{}^\top \bar{\Lambda}_{I_3} K_{I_3} + 2\lambda_2 K_{I_1}{}^\top K_{I_1} + K. \tag{29}$$

where $(u_1)_i = -2\lambda_2 r_i * y_i$, $(u_2)_i = \left(\frac{2\lambda_2}{m} \varphi_p(r_i) - \lambda_1\right) \varphi'_p(r_i) * y_i$, $(u_3)_i = \left(-\lambda_1\right) \varphi'_p(r_i) * y_i$, $\boldsymbol{r} = (r_1, ...r_m)^\top$, $\boldsymbol{\varphi_p(r)} = (\varphi_p(r_1), ...\varphi_p(r_m))^\top$, $\boldsymbol{\varphi'_p(r)} = (\varphi'_p(r_1), ...\varphi'_p(r_m))^\top$, $\Lambda = \left(\lambda_1 - \frac{2\lambda_2}{m} \boldsymbol{\varphi_p(r)}^\top \boldsymbol{e}\right) diag \quad [\varphi''_p(r_1)..., \varphi''_p(r_m)]$, $\bar{\Lambda} = \lambda_1 diag[\varphi''_p(r_1)..., \varphi''_p(r_m)]$, $q_i = (K \boldsymbol{\varphi'_p(r)})_i * y_i$, $\boldsymbol{e} = (1, ...1)^\top$, and $I_1 = \{i \in M \mid r_i > 0\}$ To improve computational efficiency, let $I_2 = \{i \in M \mid \varphi'_p(r_i) \geq \varsigma\}$, $I_3 = \{i \in M \mid \varphi''_p(r_i) \geq \varsigma\}$ for a tiny number ς like 10^{-10}. $\varphi'(r)$ and $\varphi''(r)$ are calculated as $\varphi'(r) = \frac{\min\{1, e^{pr}\}}{1 + e^{-p|r|}}$, $\varphi''(r) = \frac{p \exp(-p|r|)}{(1 + \exp(-p|r|))^2}$.

It solves Newton equation

$$\nabla^2 f(\boldsymbol{\alpha}^t) \boldsymbol{d} = -\nabla f(\boldsymbol{\alpha}^t) \tag{30}$$

to update the current solution. Let $\bar{\boldsymbol{d}}$ is the solution of (30). If the full Newton step is acceptable, then $\boldsymbol{\alpha}^{t+1} = \boldsymbol{\alpha}^t + \bar{\boldsymbol{d}}$, otherwise $\boldsymbol{\alpha}^{t+1} = \boldsymbol{\alpha}^t + \gamma \bar{\boldsymbol{d}}$, where γ is chosen by Armijo line search.

The reduced smoothing Algorithm for RA-SVCM and sRA-SVCM on large datasets is given as follows. For regular datasets, $J = M$, that is $\boldsymbol{\alpha} = \boldsymbol{\alpha}_J$.

Algorithm 1 Algorithm for RA-SVCM and sRA-SVCM

```
Input: Dataset $S = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \dots (\boldsymbol{x}_m, y_m)\}$, $\lambda_1, \lambda_2, p, \varepsilon, \varepsilon_1 > 0, t_{\max}, \beta \in (0, 0.5)$ and given an initial $\boldsymbol{\alpha}_J^0$.
Output: $\boldsymbol{\alpha}$ .
    1. Choose sub-index set $J$ from $M$ randomly, calculate reduced kernel matrix $K_{JM}$;
    2: Calculate $g^t = \nabla f(\boldsymbol{\alpha}_J^t)$, If $\|g^t\| < \varepsilon$ or $\left\|\boldsymbol{\alpha}_J^{t+1} - \boldsymbol{\alpha}_J^t\right\| < \varepsilon_1$ stop; Otherwise go to step4;
    3: Calculate $d^t$ by solving the Newton systerm $\nabla^2 f(\boldsymbol{\alpha}_J^t) d = -\nabla f(\boldsymbol{\alpha}_J^t)$;
    4: (Armijo line search) choose the largest $\gamma_t$ from the sequence $\{1, 2^{-1}, 2^{-2}, \dots\}$ such that $f(\boldsymbol{\alpha}_J^t + \gamma_t d^t) \leq f(\boldsymbol{\alpha}_J^t) + \beta \gamma_t \nabla f(\boldsymbol{\alpha}_J^t)^T d^t$;
    5: Let $\boldsymbol{\alpha}_J^{t+1} =: \boldsymbol{\alpha}_J^t + \gamma_t d^t$ and $t := t + 1$,go to step 3;
```

Setting of approximate parameters p: In order to make the Newton method working well, the approximate parameters p should be set moderately. Such as $p = 1$ at beginning, then $p := 10p$ if $\|g^t\|$ is small and repeat the algorithm until $p = 10^4$ and $\|g^t\| \leq \varepsilon$ for given ε.

Remark 1 The solution of problem (25) obtained by algorithm 1 is the globally optimal since the objective function in (25) is convex. And the solution of problem (24) obtained by algorithm 1 is almost globally optimal, because the objective function in (24) is convex in most cases. We defer the relevant proof to Appendix.

4 Differences with related methods

There are a few studies considered the effect of the moment penalization on the generalization ability of SVM-style algorithms. Zhang and Zhou [25, 26] proposed the LDM and ODM algorithm, whose idea is to optimize the margin distribution by considering the margin mean and variance simultaneously. Our method optimizes losses distribution by minimizing the mean and variance of loss function.

From the generalization error bound (23) of RA-SVCM we proposed, We can derive the statistical characteristics our methods. Under appropriate conditions on the loss l, parameter space Θ, inequality (23) shows that the generalization bound of RA-SVM can be upper bounded by the sum of three components, among which, the first term is the average of empirical loss, and the last term is a by-product which can be ignored, so the main inspiration comes from the second term. It's not difficult to find that the smaller $\mathbb{V}_m(l)$, the smaller this term, so that the tighter the bound. Thus to achieve good generalization performance, we should minimize the upper bound of loss. Hence minimizing the mean and variance of the loss can result in good generalization performance.

The generalization bound of ODM presented in the Theorem 5.1 of [26] shows that the generalization bound of ODM also can be upper bounded by the sum of three components, among which, the first term is the average of margin, and the last term is a by-product of McDiarmid inequality which can be ignored, but the second term is affected by four related parameters.

Meanwhile, the resultant objective function (6) of LDM is quite complex, and both LDM and ODM require tuning four parameters. Comparatively speaking, the objective function of our method is relatively simple, and fewer parameters need to be adjusted.

5 Empirical studies

In this section, we investigate the performance of our proposed methods using the artificial and benchmark datasets. We first introduce the experiment settings in Sect. 5.1. Then, we visualize the classifier SVM, ODM, RA-SVCM and sRA-SVCM on artificial dataset in Sect. 5.2, and then compare RA-SVCM and sRA-SVCM with standard SVM, LSSVM [11] (RRP-LSSVM [14]), SVM+ [19], SVM-2V [20] and ODM in Sect. 5.3. And the Friedman test is employed to compare the test accuracies of six methods on twenty

Table 1 Characteristics of experimental datasets and the optimal parameters

Scale	ID	Dataset	Instance	Feature	h	RA-SVCM (λ_1	λ_2)	sRA-SVCM (λ_1	λ_2)
Regular	1	Liver-disorders	145	5	2^{-1}	10^{1}	10^{1}	10^{1}	10^{1}
	2	Wine	178	13	2^{-2}	10^{-1}	10^{1}	10^{1}	10^{-1}
	3	Soner	208	60	2^{0}	10^{-1}	10^{3}	10^{1}	10^{1}
	4	Heart	270	13	2^{-6}	10^{2}	10^{-1}	10^{1}	10^{-1}
	5	Ionosphere	351	34	2^{-2}	10^{1}	10^{1}	10^{1}	10^{1}
	6	wbdc	569	30	2^{-2}	10^{1}	10^{-1}	10^{-1}	10^{1}
	7	Breast-cancer	683	10	2^{2}	10^{-1}	10^{0}	10^{-1}	10^{0}
	8	Fourclass	690	2	2^{4}	10^{1}	10^{1}	10^{-1}	10^{1}
	9	German	800	24	2^{-4}	10^{2}	10^{-1}	10^{2}	10^{-1}
	10	Vehicle	846	18	2^{-2}	10^{3}	10^{2}	10^{3}	10^{1}
	11	svmguide3	1284	22	2^{-4}	10^{1}	10^{3}	10^{1}	10^{3}
	12	svmguide1	7089	4	2^{1}	10^{-1}	10^{0}	10^{-1}	10^{0}
Large	13	USPS3	7291	256	2^{-4}	10^{0}	10^{3}	10^{0}	10^{2}
	14	USPS8	7291	256	2^{-4}	10^{0}	10^{1}	10^{0}	10^{1}
	15	Adult	32561	123	2^{-6}	10^{2}	10^{3}	10^{-1}	10^{0}
	16	Shuttle	43500	9	2^{4}	10^{2}	10^{2}	10^{2}	10^{2}
	17	MNIST3	60000	784	2^{-6}	10^{0}	10^{1}	10^{1}	10^{1}
	18	MNIST8	60000	784	2^{-6}	10^{0}	10^{1}	10^{1}	10^{1}
	19	IJCNN	49990	22	2^{0}	10^{1}	10^{0}	10^{0}	10^{1}
	20	Vechile	78823	100	2^{-3}	10^{4}	10^{4}	10^{4}	10^{5}

Note: h, λ_1 and λ_2 indicate kernel spread parameters, coefficients of mean and variance respectively

datasets in Sect. 5.4. In addition, we also study the loss distribution and relevant results produced by sRA-SVCM, SVM, LSSVM (RRP-LSSVM) and ODM in Sect. 5.5. Finally, the sensitivity of parameters is analyzed in Sect. 5.6, and the time cost of four method are compared in 5.7.

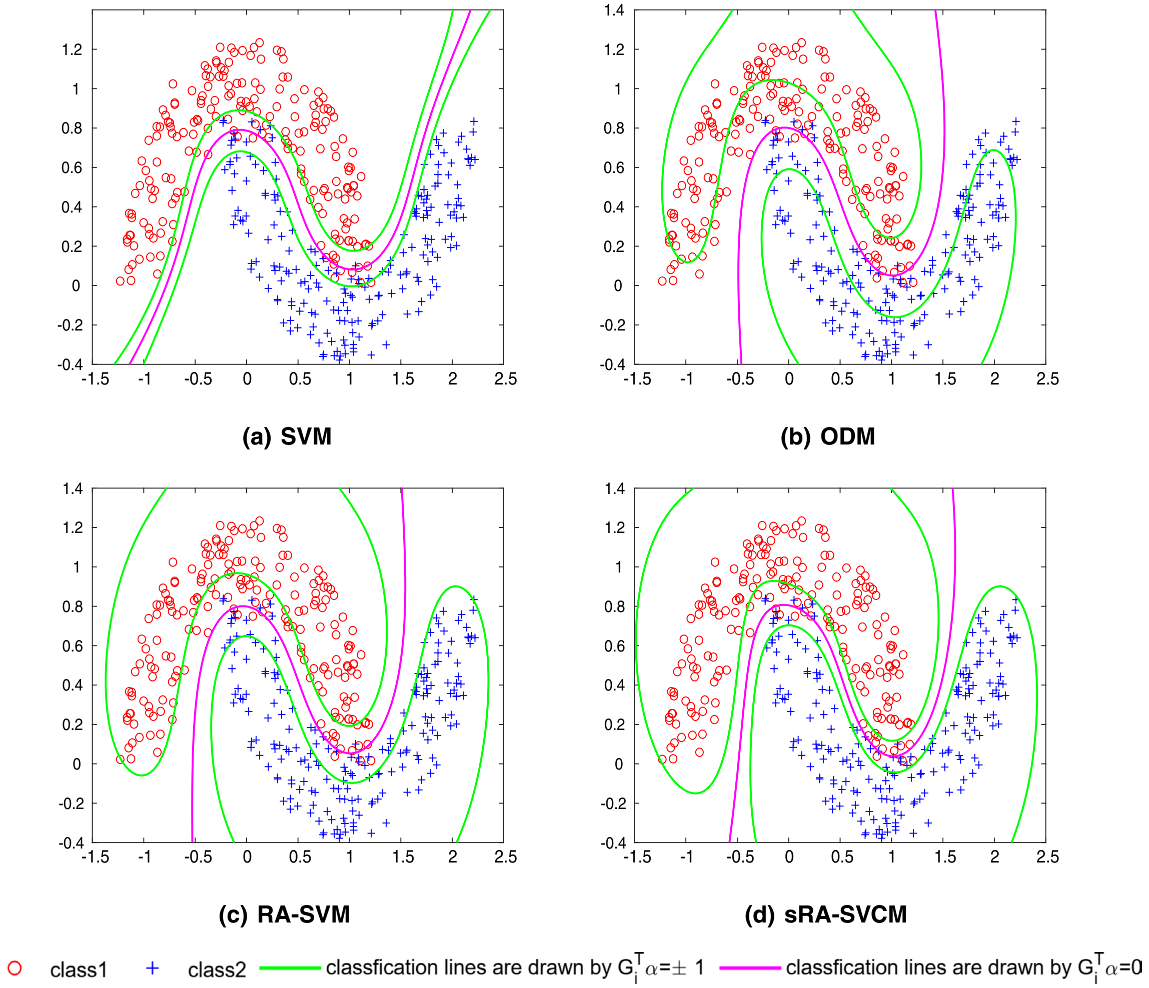
5.1 Experimental setup

We evaluate the effectiveness of our models on a artificial dataset, twelve regular scale datasets and eight large scale datasets, including both UCI data sets and real-word data sets like KDD2010. Table 1 summarizes the statistics of these datasets. The size of the datasets ranges from 145 to 78823, and the dimensionality ranges from 2 to 784. All features are normalized into the interval [0,1]. Meanwhile, the optimal parameters are listed in table, and the abbreviations in table are explained at the bottom of the table. The Gaussian kernel function $K(x_1, x_2) = \exp(-h\|x_1 - x_2\|^2)$ is used for all datasets, kernel spread parameters h, regularization parameters λ_1 ,λ_2 are roughly chosen by 5-fold cross validation within $h \in \{2^{-6}, 2^{-5}......, 2^{5}, 2^{6}\}$ and $\lambda_1, \lambda_2 \in \{10^{6}, 10^{5}......10^{-6}\}$.

The regular scale datasets and *Adult*, *IJCNN* of larger datasets are original binary classification problems, the other five datasets are muti-class datasets. Specifically, the task of separating class 3 from the rest is trained on *Vechile* dataset. For *Shuttle* data set, a binary classification problem is solved to separate class 1 from the rest. And for *MNIST* and *USPS* dataset, here two binary classification problems are solved to separate class 3 from the rest and separate class 8 from the rest. For the datasets with partition of training sets and testing sets, we can adopt the original training sets and testing sets directly. And for the data sets without partition of training set and testing set, we can select eighty percent of the instances randomly as training data, and use the rest as testing data.

For standard SVM, LSSVM, SVM+ and SVM-2V, the regularization parameter λ and Kernel spread parameter h are selected by 5-fold cross validation from the set of $\{10^{-6}, 10^{-5} \cdots 10^{6}\}$ and $\{2^{-6}, 2^{-5} \cdots 2^{6}\}$ respectively, and for RRP-LSSVM, 1000 of the training data is randomly selected as the working set for larger datasets. For ODM, the regularization parameter λ_1 are selected by 5-fold cross validation from the set of $\{10^{-6}, 10^{-5}.....10^{6}\}$, while the parameters D and μ is selected from the set of $\{0.2, 0.4, 0.6, 0.8\}$. For SVM-2V, other parameters are set as [20]. Experiments are repeated for 30 times, and the average accuracies as well as the standard deviations are recorded.

All the experiments are carried out on a desktop PC with Intel(R) Core(TM)i7-7700 CPU (3.60 GHz) and 16GB RAM under the MATLAB 2019b programming environment.



(a) SVM

(b) ODM

(c) RA-SVM

(d) sRA-SVCM

Fig. 2 Plots for comparing the classification boundaries of SVM, ODM, RA-SVCM and SRA-SVCM on the two moon dataset. For this dataset, the test accuracies of these four algorithms are 97.00%, 97.50%, 97.75% and 97.50%

5.2 Experimental results with artificial data

We visualize the classification hyperplanes determined by SVM (7), ODM, RA-SVCM and the sRA-SVCM model (10) on a two-moon dataset with 800 samples (400 for training, 400 for testing) and $d = 2$ features. Figure 2 shows the experimental results.

From the comparisons of Fig. 2a–d, it can be seen that for the two moon dataset, the classification boundaries of ODM, RA-SVCM, and sRA-SVCM are relatively similar, and the classification boundaries of these three methods are obviously better than that of standard SVM. Meanwhile, the classification accuracies of these three methods are higher than that of standard SVM. Compared with standard SVM, the margin of RA-SVCM that we proposed is wider than that of SVM, and the classification confidence of our methods are obviously higher than that of SVM on Data from the end of the crescent moon. Compared with ODM, the margin of RA-SVCM that we proposed is narrower than that of ODM but the number of support vector is smaller than ODM. We can also find that the classification performance of RA-SVCM and sRA-SVCM is similar. It can be concluded that our method has good generalization performance.

5.3 Experimental results with regular scale and large benchmark Datasets

According to the experimental setup in Sect. 4.1, experiments were carried out on the twenty datasets above, the results summarized in Table 2 (the experimental result of the SVM-2V and SVM+ on the eight scale data is missing because the quadratic programming is difficult to deal with large kernel matrix).

As can be seen, the overall performance of our models are superior to the other compared methods. According to the test accuracy, these twenty datasets can be divided into Two levels: The datasets which test accuracy less than 95% are "hard" datasets; The rest are belonging to "easy" datasets.

Table 2 Accuracy (mean ± td) comparison on regular and large scale data sets

Scale	Data set	SVM	LSSVM	SVM+	SVM-2v	ODM	RA-SVCM	sRA-SVCM
Regular	Liver disorders	73.10 (1.20)	74.25 (6.42)	75.52 (5.26)	75.86 (7.62)	75.40 (6.70)	**76.55**(7.10)	**76.55**(6.20)
	Wine	96.72 (2.85)	96.57 (2.80)	98.33 (1.94)	**98.89**(1.43)	98.33 (2.99)	98.52 (1.96)	98.61 (1.59)
	Soner	87.94 (4.00)	88.17 (5.88)	90.48 (6.04)	90.05 (6.33)	89.44 (4.23)	**91.19**(4.87)	90.95 (3.38)
	Heart	84.26 (4.96)	84.01 (4.63)	81.48 (3.08)	83.70 (1.46)	83.33 (5.24)	**85.93**(3.90)	85.19 (3.62)
	Ionosphere	91.43 (1.78)	89.29 (2.70)	91.86 (3.50)	92.00 (3.10)	92.19 (3.04)	93.00 (2.78)	**93.43**(2.81)
	wbdc	98.33 (0.77)	97.57 (1.12)	98.25 (1.32)	97.63 (1.37)	98.16 (0.97)	**98.51**(0.55)	98.42 (1.17)
	Breast-cancer	97.35 (1.35)	97.57 (1.06)	97.59 (0.91)	97.51 (1.38)	97.62 (1.21)	**98.25**(1.01)	98.00 (0.51)
	Fourclass	99.94 (0.18)	**100.00**(0.25)	**100.00**(0.00)	**100.00**(0.00)	**100.00**(0.00)	100.00 (0.00)	**100.00**(0.00)
	German	71.00 (0.00)	70.00 (0.01)	**71.02**(0.03)	71.00 (0.01)	68.50 (0.00)	71.00 (0.01)	71.00 (0.00)
	Vehicle	84.99 (2.78)	83.55 (1.99)	84.38 (2.91)	84.78 (1.23)	85.25 (2.53)	86.27 (2.79)	**86.57**(2.62)
	svmguide3	**80.16**(0.72)	78.60 (0.33)	79.46 (0.61)	79.38 (0.54)	79.07 (0.87)	**80.16**(0.41)	**80.16**(0.52)
	svmguide1	96.04 (0.59)	95.90 (0.43)	94.66 (2.16)	95.00 (0.63)	96.10 (0.79)	**96.15**(0.70)	**96.15**(0.62)
Scale	Data set	SVM	RRP-LSSVM	SVM+	SVM-2v	ODM	RA-SVM	sRA-SVM
Lager	USPS3	98.60 (0.05)	98.38 (0.00)	–	–	**98.72**(0.06)	0.9860 (0.04)	98.60 (0.04)
	USPS8	99.34 (0.08)	99.05 (0.00)	–	–	**99.38**(0.08)	**99.38**(0.08)	**99.38**(0.08)
	Adult	85.05 (0.01)	85.07 (0.00)	–	–	85.25 (0.07)	**85.34** (0.06)	**85.34**(0.05)
	Shuttle	99.88 (0.01)	99.83 (0.00)	–	–	99.89 (0.01)	**99.92**(0.01)	**99.92**(0.01)
	minist3	99.04 (0.06)	**99.10** (0.00)	–	–	99.05 (0.05)	99.09 (0.05)	**99.10**(0.05)
	minist8	99.29 (0.04)	99.24 (0.00)	–	–	99.31 (0.07)	**99.32**(0.05)	**99.32**(0.05)
	ijcnn1	98.72 (0.01)	96.60 (0.00)	–	–	98.58 (0.24)	98.75 (0.13)	**98.78**(0.05)
	Vechile	87.84 (0.04)	88.04 (0.00)	–	–	88.00 (0.07)	88.04 (0.04)	**88.05**(0.04)

All the results are the mean of 30 random trials, the best values are in bold

As showing in Table 2, For "easy" datasets, the test accuracy of our methods can do up to 1.95% better than the standard SVM; For "hard" datasets, the test accuracy of our methods can do up to 4.71% better than the standard SVM. This indicates that our methods have a better generalization ability than the standard SVM, especially for the "hard" datasets.

Meanwhile, it can be seen from the experimental results that sRA-SVCM and RA-SVCM behave almost similarly. For simplicity, the sRA-SVCM will be used to compare with other models.

5.4 Statistical comparisons by friedman test

In order to evaluate multiple methods systematically, the Friedman test [40] is employed to compare the test accuracies of the five methods over the 20 benchmark datasets. For the different datasets, Friedman test at significance level $\alpha = 0.05$ rejects the null hypothesis of equal performance, which leads to the use of post-hoc tests to find out which algorithms are actually different. The Nemenyi test is used to further distinguish different methods. Specifically, Nemenyi test is used where the performance of two algorithms is significantly different if their average ranks over all datasets differ by at least one critical difference

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6M}}, \tag{31}$$

where critical values q_α are based on the studentized range statistic, K is the number of comparison algorithms, and M is the number of datasets.

Take Fig. 3a as an example, for the five methods on twelve datasets, according to

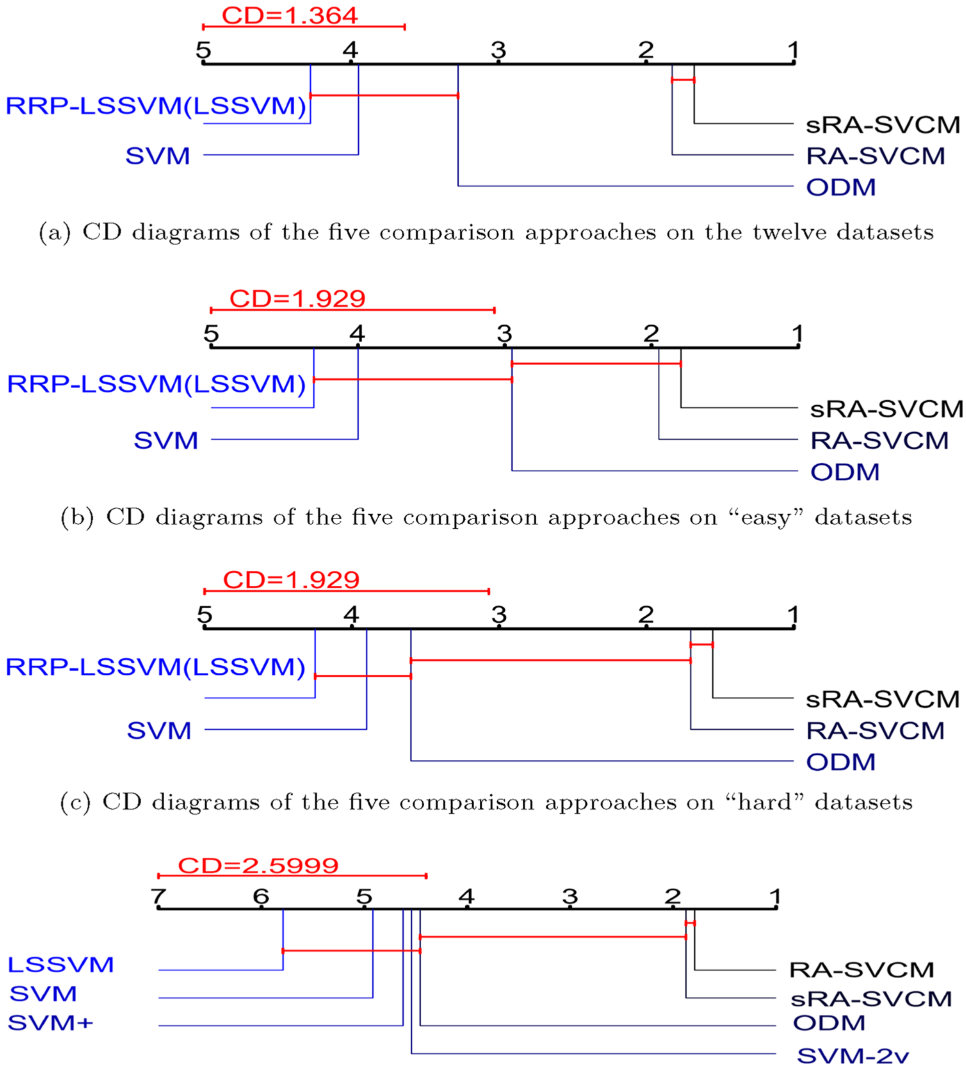
$$\tau_{\chi^2} = \frac{12M}{k(k+1)}\left(\sum_{\mathrm{i}=1}^{k} o_i^2 - \frac{k(k+1)^2}{4}\right) \tag{32}$$

and

$$\tau_F = \frac{(M-1)\tau_{\chi^2}}{M(k-1)-\tau_{\chi^2}}, \tag{33}$$

the friedman statistic $\tau_F = 25.60$, which at significance level $\alpha = 0.05$ rejects the null hypothesis of equal performance, where o_i is the average ordinal number of the ith algorithm, and k is the number of algorithms. Then, the critical difference ($CD = 1.3640$) is calculated by (31). If the difference between the average ranks of the two algorithms exceeds the critical value 1.3640, the assumption that "the

Fig. 3 CD diagrams of the comparison approaches on the certain datasets. Groups of methods that are not significantly different according to Nemenyi test are connected with a red line



(a) CD diagrams of the five comparison approaches on the twelve datasets

(b) CD diagrams of the five comparison approaches on "easy" datasets

(c) CD diagrams of the five comparison approaches on "hard" datasets

(d) CD diagrams of the seven comparison approaches on regular datasets

two algorithms have the same performance" is rejected with corresponding confidence.

Figure 3 illustrates the CD diagrams for the comparison methods on the twenty benchmark datasets, where the average rank of each comparing method is marked along the axis. The axis is turned so that the best ranks are to the right. Groups of methods that are not significantly different according to Nemenyi test are connected with a red line. The critical difference is also shown above the axis in each subfigure.

As can be seen from Fig. 3, our methods achieve the statistically superior performance on the whole twenty datasets. Our two methods are not significantly different from ODM on "easy" datasets, but are significantly superior to the other methods. The sRA-SVCM are significantly different from ODM, SVM and R-LSSVM on"hard" datasets, and our two methods are significantly different from SVM, LSSVM, SVM+ and SVM-2v on regular datasets. Meanwhile, we can also see that the generalization performance of our method is best on the whole datasets, and our two methods (sRA-SVCM and sRA-SVCM) are not significantly different. So for simplicity, the sRA-SVCM will be used to compare with other models in the following experiments.

5.5 Specific results and loss distribution on four datasets

In this section, the experiments of sRA-SVCM, standard SVM, ODM and LSSVM (RRP-LSSVM) are performed on *svmguide1*, *Adult*, *Vechile* and *IJCNN* datasets (the optimal parameters of four models are used). Some specific results of these experiments are listed in Table 3.

As can be seen from Table 3, the training accuracy of our model are not always the best, but the test accuracy is always the largest. This is because the mean of loss of sRA-SVCM is slightly larger than that of SVM, but the variance

Table 3 Experimental results on four representative datasets

	Method	μ	σ	TRA (%)	TEA (%)	AUC	F1-score	$S_{r<=0}$	S_{0-1}	$S_{r>1}$
svmguide1	SVM	0.0853	0.1170	96.96	96.05	0.9956	0.9616	2755	240	94
	LSSVM	0.2229	0.0955	96.89	95.85	0.9880	0.9597	1179	1814	97
	ODM	0.1696	0.0907	97.15	96.10	0.9960	0.9621	1747	1254	88
	sRA-SVCM	0.0887	0.0917	**97.35**	**96.15**	**0.9961**	**0.9626**	2656	351	82
Adult	SVM	0.3422	0.4111	84.24	84.87	0.9008	0.6377	20914	6675	4972
	RRP-LSSVM	0.4247	0.2243	85.56	85.23	0.9012	0.6529	7071	20788	4702
	ODM	0.5094	0.1722	**85.66**	85.22	0.9038	0.6567	3972	23920	4669
	sRA-SVC	0.4042	0.2440	85.38	**85.38**	**0.9044**	**0.6581**	13095	14707	4759
IJCNN	SVM	0.0392	0.0517	98.64	98.81	0.9949	0.9366	46963	2348	679
	RRP-LSSVM	0.1298	0.0492	98.63	98.18	0.9911	0.9006	20070	29237	683
	ODM	0.0396	0.0302	**99.12**	98.23	0.9940	0.9101	45540	4012	438
	sRA-SVCM	0.0480	0.0389	98.80	**98.84**	**0.9957**	**0.9388**	44762	4630	598
Vechile	SVM	0.2783	0.4097	87.62	87.88	0.9226	0.8712	56887	12178	9758
	RRP-LSSVM	0.3953	0.2243	87.69	87.96	0.9271	0.8738	12197	56923	9703
	ODM	0.4588	0.1817	87.86	87.99	**0.9275**	0.8737	12056	57195	9572
	sRA-SVCM	0.3604	0.2465	**87.92**	**88.13**	**0.9275**	**0.8751**	25171	44132	9520

Here μ, σ, “TRA”, and “TEA” are indicate “Mean of loss”,“Varivance of loss”, “Train accuracy ”and “Test accuracy ” respectively. $S_{r<0}$, S_{0-1} and $S_{r>1}$ are the numbers of samples in the interval $[r_{\min}, 0)$, (0, 1] and $(1, r_{\max}]$ for the loss

of loss of sRA-SVCM is slightly less than that of SVM. The differences bring good results that the test accuracy of sRA-SVCM have been improved than the standard SVM. ODM optimizes the distribution of margin, to a certain extent, and it also optimizes the distribution of loss. SVM does not care about the variance of the loss, so the variances of loss are lager than RA-SVCM and ODM. For the LSSVM, the loss of most points are concentrated between 0 and 1, which makes too many points near the classification hyperplane. In this case, their generalization performance will be slightly lower than sRA-SVCM. Meanwhile, the AUC and F1-score of RA-SVCM are superior to SVM. The above show that our method have a good generalization ability and better performance.

Figure 4 plots the positive loss distribution of sRA-SVCM and standard SVM on training sets of four representative datasets. It can be seen that the positive loss of SVM is relatively scattered and the maximum value larger than our models. That of RA-SVCM is relatively concentrated, and the number of sRA-SVCM between 0 and 1 is higher than SVM. And the Fig. 5 plots the corresponding results of r on test sets of the four datasets. It is obvious that the test error of our method is smaller than that of SVM, and the maximum error is much smaller than the standard SVM. So that our model has a better classification ability on the test sets. Namely, our model has a better generalization performance than standard SVM. A reasonable interpretation is that there have a suitable loss distribution of the new model on the training set, and thus it get a better generalization ability on test sets.

5.6 Sensitivity analysis of parameter of RA-SVCM

There are two tuned parameters in the proposed model, such as λ_1 and λ_2, which are used to balance the importance of the corresponding terms. The first term is used to make the points generated by the model closer to the sample datas. The second term guarantees the stability of the entire sample data in the model.

To analyze the sensitivity of parameters in the proposed method, we first define a candidate set where the optimal parameter located for these parameters. We have performed the proposed model 20 times with the parameters in candidate set and report the mean classification accuracy. From the Fig. 6, it is obvious that the average test accuracies are almost steady with respect to different values of parameter λ_1 and λ_2, which indicates that those parameters do not require careful tuning, and a broad range of λ_1 and λ_2 lead to satisfactory performance.

Next, we fix parameter λ_2 and select the seven candidate parameters of λ_1 to compare with SVM-H and SVM-SH (standard SVM with hinge loss and squared hinge loss) on datasets above. The Fig. 7 shows the relationships of the average test accuracy and mean parameters λ_1 of sRA-SVCM, SVM-H and SVM-SH. And the Fig. 7 represents the average test accuracies of sRA-SVCM, SVM-H and SVM-SH by red line, magenta line, and green line respectively. It

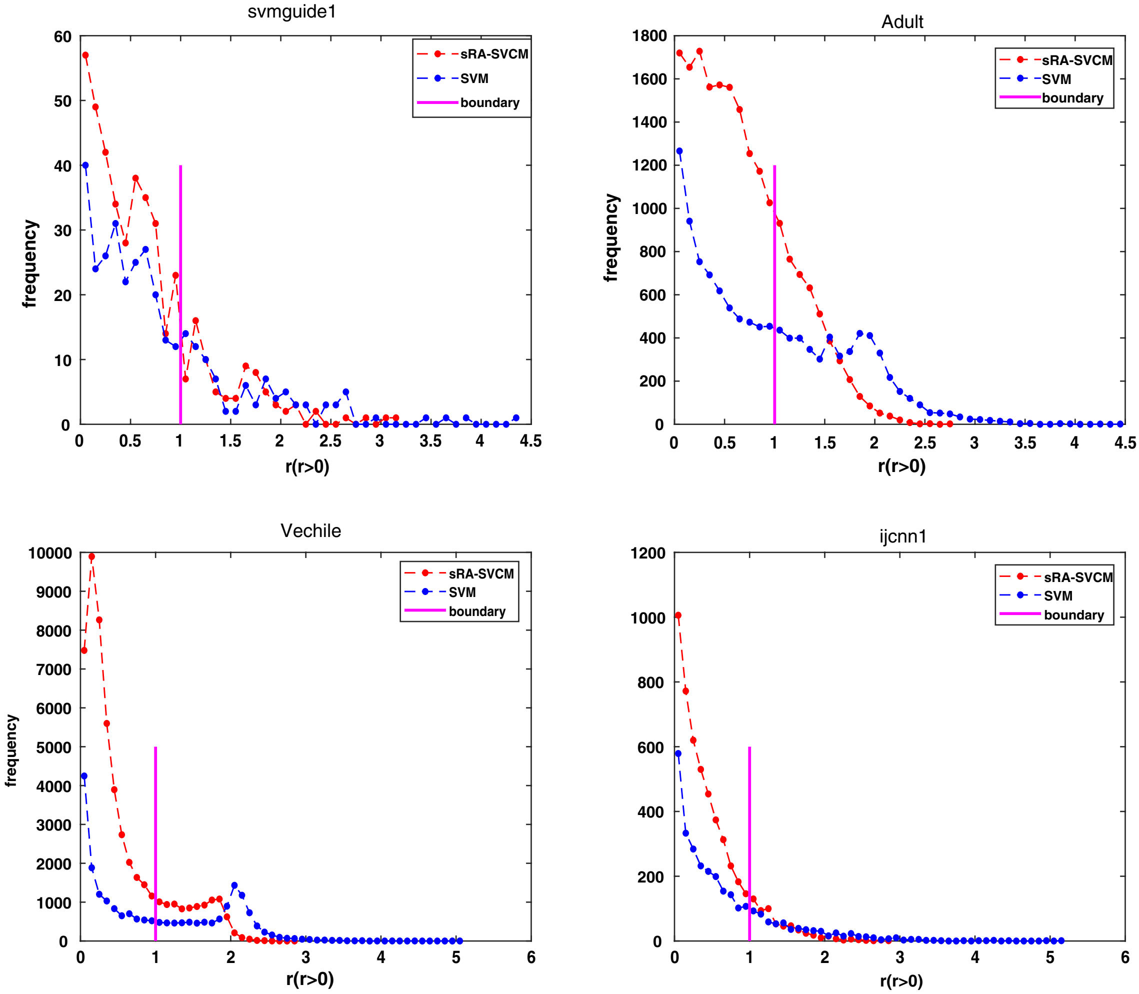


Fig. 4 Plots for comparing positive loss distribution on training datasets of *svmguide1*, *Adult*,*Vechile* and *IJCNN*

is obvious that the magenta and green lines fluctuate greatly with the change of the parameters λ_1, but the red lines are relatively stable, and the red lines are generally above the magenta and green lines. This indicates that compared with SVM-H and SVM-SH, the sRA-SVCM achieve a better generalization performance, and the sRA-SVCM is not sensitive to the parameter λ_1. The sensitivity analysis to λ_1 reveals that this conclusion is stable to reasonable pertubations of λ_1.

5.7 Time cost

We compare the cost of our methods with SVM and ODM on six large scale datasets(RRP-LSSVM, SVM+ and SVM-2v can be solved without iterative methods). All the experiments are carried out on a desktop PC with Intel(R) Core(TM)i7-7700 CPU (3.60 GHz) and 16GB RAM under the MATLAB 2019b programming environment. The average CPU time (in seconds) on each dataset is show in Fig. 8. The Newton algorithm is used to solve these four models. It can be seen that, except for the vechile datasets, our methods are faster than SVM and slightly faster than ODM. On vechile datasets, RA-SVCM is slightly slower than ODM, but sRA-SVCM is still faster than ODM. This show that our methods are computationally efficient.

6 Conclusion

Recent theoretical results suggested that the distribution of loss, rather than only mean of loss, is more crucial to the generalization performance. In this paper, based on the empirical Bernstein inequality, we propose a novel method, named Risk-Averse support vector classifier machine (RA-SVCM), which tries to optimize the loss distribution by considering the mean and variance of loss. Our models are general learning approach which can be used in any place where SVM can be applied. Comprehensive experiments on

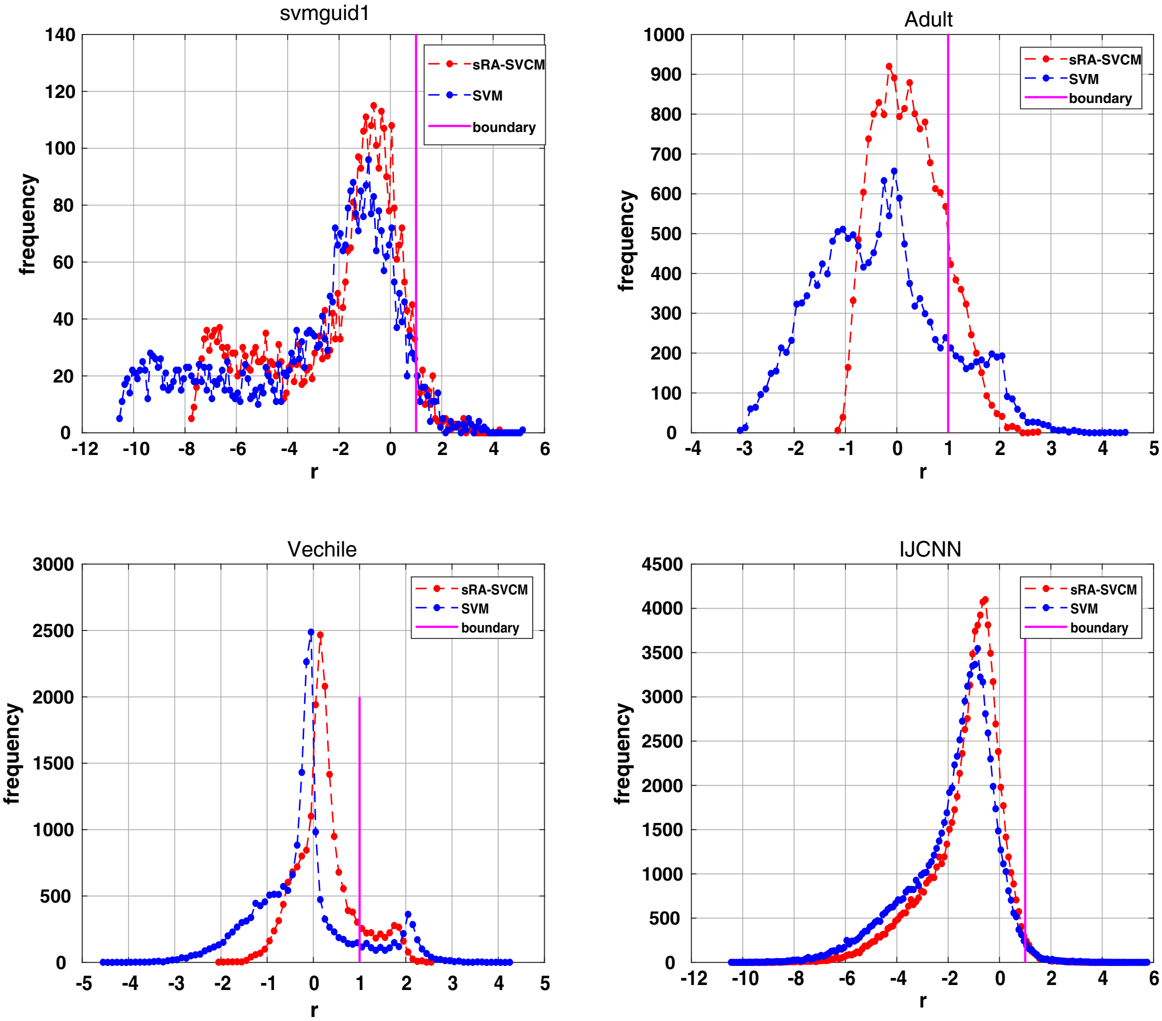


Fig. 5 Plots for comparing *r* distribution on test datasets of *svmguide1*, *Adult*, *Vechile* and *IJCNN* datasets

the artificial and benchmark datasets validate the superiority of our method to other classical models.

In the future, it will be interesting to further investigate the application of mean-variance minimization to other particular loss functions (Such as ramp loss, squares hinge loss, etc). And this raise a pressing issue to find an efficient implementation, which can deal with the case that the sample variance penalization is non-convex. In addition, it is necessary to automatically estimate the parameter λ_2 without cross-validation in order to make RA-SVCM free from additional parameters. And another line of future research is to refine these generalized boundaries with additional theoretical work.

Appendix A Proof of remark 1

Theorem 2 *Under the condition of Theorem 1, the objective function $f_1(\boldsymbol{\alpha})$ in (24) is convex if*

$$\frac{\lambda_1}{\lambda_2} \geq 2\left(c + \frac{1}{p}\right). \tag{A1}$$

Proof Here, let $g_i(\boldsymbol{\alpha}) = \frac{1}{p}\log(1+e^{pr_i}) = \max\{\boldsymbol{r}_i, 0\} + \frac{1}{p}\log(1+e^{-p|\boldsymbol{r}_i|})$, the objective function of (24) can be written as

$$f_1(\boldsymbol{\alpha}) = \lambda_1 \sum_{i=1}^{m} g_i(\boldsymbol{\alpha}) + \lambda_2 g^{\top}(\boldsymbol{\alpha}) Q g(\boldsymbol{\alpha}) + \frac{1}{2}\boldsymbol{\alpha}^{\top} K \boldsymbol{\alpha} \tag{A2}$$

The Hessian matrix of $f_1(\boldsymbol{\alpha})$ is:

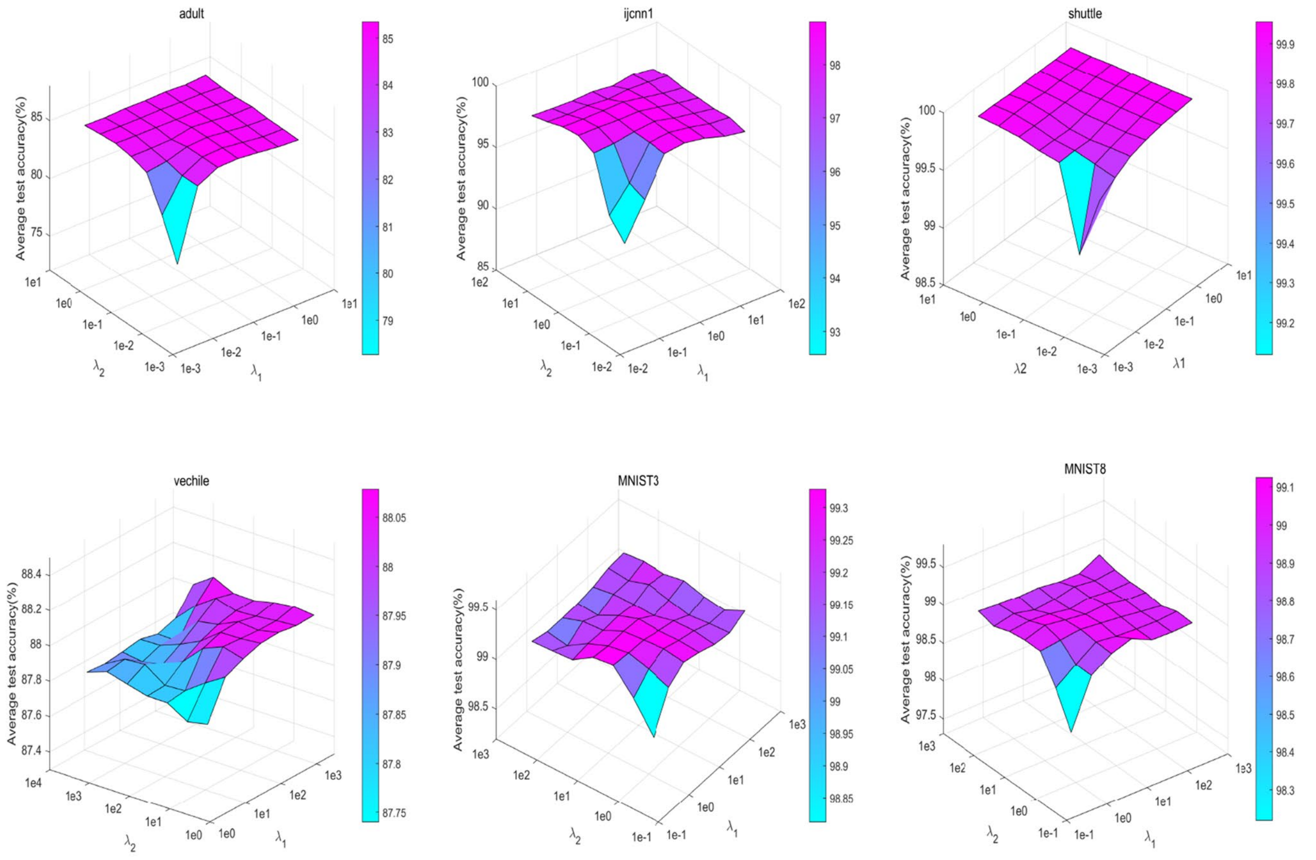


Fig. 6 Plots for relationships of the average test accuracies (%) and different combinations of parameters on the six larger datasets

$$
\begin{aligned}
\nabla^2 f_1(\boldsymbol{\alpha}) =&\lambda_1 \sum_{i=1}^{m} \nabla^2 g_i(\boldsymbol{\alpha}) + 2\lambda_2 \sum_{i=1}^{m} \boldsymbol{\delta}_i \nabla^2 g_i(\boldsymbol{\alpha}) \\
&+ 2\lambda_2 \nabla g(\boldsymbol{\alpha}) Q \nabla g^{\top}(\boldsymbol{\alpha}) + K \\
=& \sum_{i=1}^{m} (\lambda_1 + 2\lambda_2 \boldsymbol{\delta}_i) \nabla^2 g_i(\boldsymbol{\alpha}) \\
&+ 2\lambda_2 \nabla g(\boldsymbol{\alpha}) Q \nabla g^{\top}(\boldsymbol{\alpha}) + K
\end{aligned}
\tag{A3}
$$

where $Q = \boldsymbol{I} - \frac{1}{m}\boldsymbol{e}^{\top}\boldsymbol{e}$, $\boldsymbol{\sigma} = Qg(\boldsymbol{\alpha})$, and

$$
\begin{aligned}
\boldsymbol{\delta}_i =& g_i(\boldsymbol{\alpha}) - \frac{1}{m} \sum_{i=1}^{m} e^T g(\boldsymbol{\alpha}) \\
=& \max\{\boldsymbol{r}_i, 0\} + \frac{1}{p} \log(1 + e^{-p|\boldsymbol{r}_i|}) - \frac{1}{m} \sum_{j=1}^{m} \max\{\boldsymbol{r}_j, 0\} \\
&+ \frac{1}{p} \log(1 + e^{-p|\boldsymbol{r}_j|}) \\
\geq& - \frac{1}{m} \sum_{j=1}^{m} \max\{\boldsymbol{r}_j, 0\} - \frac{1}{mp} \sum_{j=1}^{m} \log(1 + e^{-p|\boldsymbol{r}_j|}) \\
\geq& - c - \frac{1}{mp} \sum_{j=1}^{m} \log(1 + e^{-p(1+M)}) \\
\geq& - c - \frac{\log 2}{p} \\
\geq& - c - \frac{1}{p}.
\end{aligned}
\tag{A4}
$$

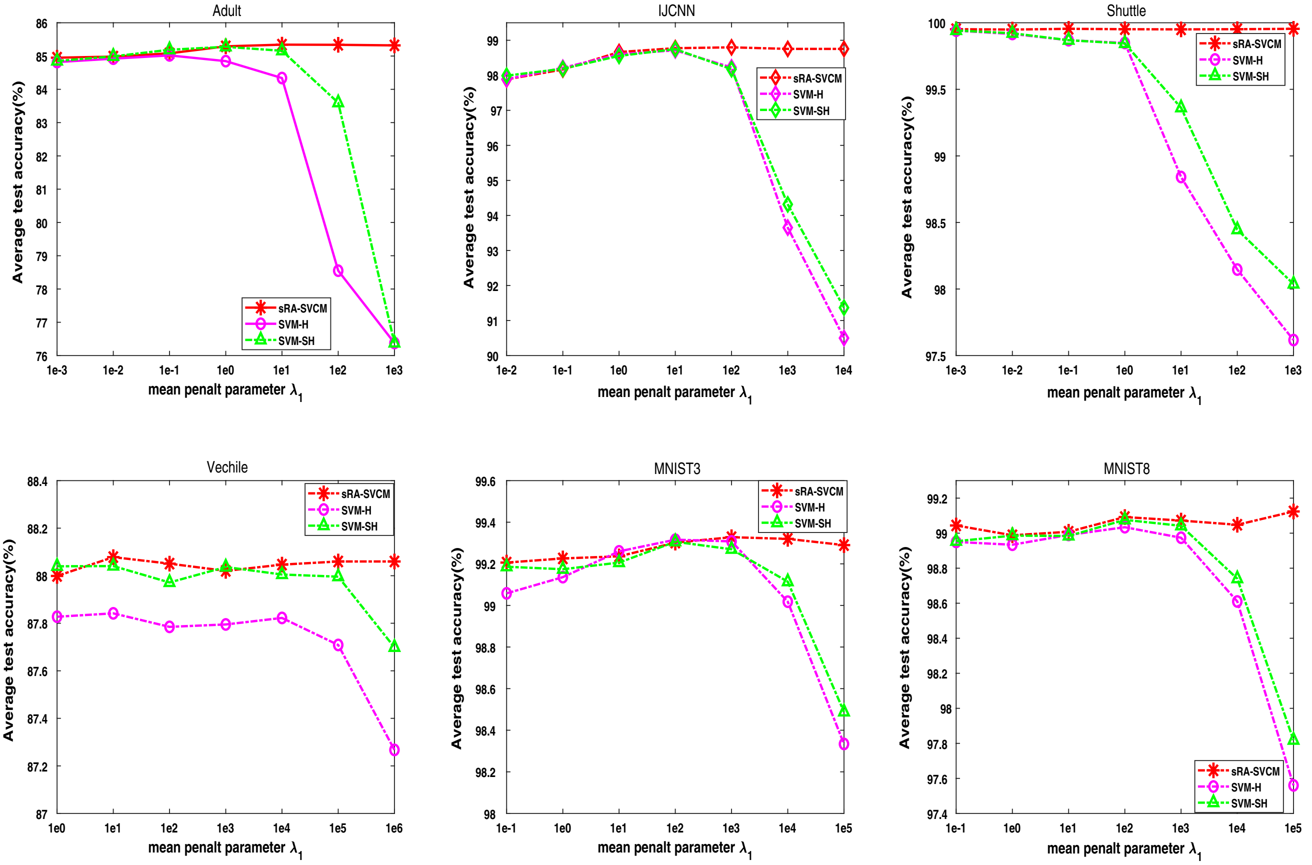


Fig. 7 Plots for comparing average test accuracies on sRA-SVCM, SVM-H, and SVM-SH when the five different mean penalty parameters λ_1 are chosen on six large datasets. These seven mean penalty parameters are in interval which the optimal parameter is located, and the parameter λ_2 are fixed

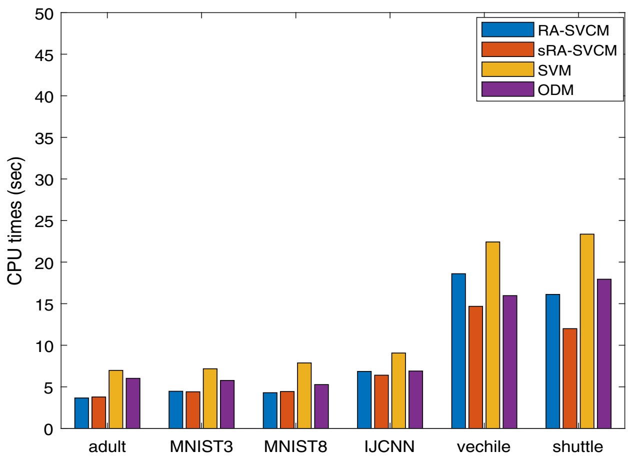


Fig. 8 CPU time on the large scale datasets

To prove that the objective function in Eq.(A2) is convex, it suffices to show that $\nabla^2 f_1(\boldsymbol{\alpha}) \geq 0$ for every $\boldsymbol{\alpha}$. It's obvious that $2\lambda_2 \nabla g(\boldsymbol{\alpha}) Q \nabla g^T(\boldsymbol{\alpha}) + K \geq 0$. That is, we need to prove that $\sum_{i=1}^{m} (\lambda_1 + 2\lambda_2 \boldsymbol{\delta}_i) \nabla^2 g_i(\boldsymbol{\alpha}) \geq 0$. Thus, let $\boldsymbol{\mu}_i = \lambda_1 + 2\lambda_2 \boldsymbol{\delta}_i$, then we need to prove $\boldsymbol{\mu}_i \geq 0$, based on (A4), we get $\frac{\lambda_1}{\lambda_2} \geq 2(c + \frac{1}{p})$. That is, when $\frac{\lambda_1}{\lambda_2} \geq 2(c + \frac{1}{p})$, for each $\boldsymbol{\alpha} \in R^m$, we have $\nabla^2 f_1(\boldsymbol{\alpha}) \geq 0$. Therefore, the objective function $f_1(\boldsymbol{\alpha})$ in (24) is convex. □

It is obvious that the objective function in Eq.(25) is convex since $\varphi_p(r)$ and $(\varphi_p(r))^2$ are convex functions. When $\frac{\lambda_1}{\lambda_2} \geq 2(c + \frac{1}{p})$, the solution of problem (24) obtained by algorithm 1 is globally optimal based on the Theorem 2. In fact, we have rarely encountered non-convergence in a large number of experiments. Of course we can also make a simple rule for selecting a optimal superparameter that satisfies the conditions given above.

Acknowledgements This work was supported by the National Natural Science Foundation of China under Grants 61772020.

References

1. Cherkassky V (1997) The nature of statistical learning theory. IEEE Trans Neural Netw 8(6):1564–1564. https://doi.org/10.1109/TNN.1997.641482
2. Vapnik VN (1999) An overview of statistical learning theory. IEEE Trans Neural Netw 10(5):988–999. https://doi.org/10.1109/72.788640
3. Osuna E, Freund R, Girosi F (2000) Training support vector machines: an application to face detection. In: IEEE Computer Society Conference on Computer Vision & Pattern Recognition, pp 130–136. https://doi.org/10.1109/CVPR.1997.609310
4. Cheng Y, Fu L, Luo P, Ye Q, Liu F, Zhu W (2020) Multi-view generalized support vector machine via mining the inherent relationship between views with applications to face and fire smoke recognition. Knowledge-Based Syst 210:106488. https://doi.org/10.1016/j.knosys.2020.106488
5. Olatunji SO (2019) Improved email spam detection model based on support vector machines. Neural Comput Appl 31(3):691–699. https://doi.org/10.1007/s00521-017-3100-y
6. Cun YL, Boser B, Denker JS, Henderson D, Jackel LD (1990) Handwritten digit recognition with a back-propagation network. In: Advances in Neural Information Processing Systems, pp 396–404 . https://dl.acm.org/doi/10.5555/109230.109279
7. Yadav A, Singh A, Dutta MK, Travieso CM (2020) Machine learning-based classification of cardiac diseases from PCG recorded heart sounds. Neural Comput Appl 32(28):17843–17856. https://doi.org/10.1007/s00521-019-04547-5
8. Yang L, Xu Z (2019) Feature extraction by PCA and diagnosis of breast tumors using svm with de-based parameter tuning. Int J Mach Learn Cybern 10(3):591–601. https://doi.org/10.1007/s13042-017-0741-1
9. Le DN, Parvathy VS, Gupta D, Khanna A, Rodrigues J, Shankar K (2021) Iot enabled depthwise separable convolution neural network with deep support vector machine for covid-19 diagnosis and classification. Int J Mach Learn Cybern. https://doi.org/10.1007/s13042-020-01248-7
10. Yu D, Xu Z, Wang X (2020) Bibliometric analysis of support vector machines research trend: a case study in china. Int J Mach Learn Cybern 11(3):715–728. https://doi.org/10.1007/s13042-019-01028-y
11. Suykens Vandewalle J (1999) Least square support vector machine classifiers. Neural Process Lett 9(3):293–300. https://doi.org/10.1023/A:1018628609742
12. Du JZ, Lu WG, Wu XH, Dong JY, Zuo WM (2018) L-SVM: a radius-margin-based svm algorithm with logdet regularization. Expert Syst Appl 102:113–125. https://doi.org/10.1016/j.eswa.2018.02.006
13. Vitt CA, Dentcheva D, Xiong H (2019) Risk-averse classification. Ann Operat Res 3:1–29. https://doi.org/10.1007/s10479-019-03344-6
14. Zhou S (2015) Sparse LSSVM in primal using Cholesky factorization for large-scale problems. IEEE Trans Neural Netw Learn Syst 27(4):783–795. https://doi.org/10.1109/TNNLS.2015.2424684
15. Khemchandani R, Chandra S et al (2007) Twin support vector machines for pattern classification. IEEE Trans Pattern Anal Mach Intell 29(5):905–910. https://doi.org/10.1109/TPAMI.2007.1068
16. Yan H, Ye Q, Zhang T, Yu D-J, Yuan X, Xu Y, Fu L (2018) Least squares twin bounded support vector machines based on l1-norm distance metric for classification. Pattern Recogn 74:434–447. https://doi.org/10.1016/j.patcog.2017.09.035
17. Richhariya B, Tanveer M (2020) A reduced universum twin support vector machine for class imbalance learning. Pattern Recogn 102:107150. https://doi.org/10.1016/j.patcog.2019.107150
18. Vapnik V, Vashist A (2009) A new learning paradigm: learning using privileged information. Neural Netw 22(5–6):544–557. https://doi.org/10.1016/j.neunet.2009.06.042
19. Gammerman A, Vovk V, Papadopoulos H (2015) Statistical learning and data sciences. In: Third International Symposium, SLDS, vol 9047, pp 20–23
20. Tang J, Tian Y, Zhang P, Liu X (2017) Multiview privileged support vector machines. IEEE Trans Neural Netw Learn Syst 29(8):3463–3477. https://doi.org/10.1109/TNNLS.2017.2728139
21. Cheng Y, Yin H, Ye Q, Huang P, Fu L, Yang Z, Tian Y (2020) Improved multi-view GEPSVM via inter-view difference maximization and intra-view agreement minimization. Neural Netw 125:313–329. https://doi.org/10.1016/j.neunet.2020.02.002
22. Ye Q, Huang P, Zhang Z, Zheng Y, Fu L, Yang W (2021) Multiview learning with robust double-sided twin svm. IEEE Trans Cybern 60:1–14. https://doi.org/10.1109/TCYB.2021.3088519
23. Garg A, Dan R (2003) Margin distribution and learning algorithms. In: Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003), August 21–24, 2003, Washington, DC, USA, pp 210–217
24. Lu X, Liu W, Zhou C, Huang M (2017) Robust least-squares support vector machine with minimization of mean and variance of modeling error. IEEE Trans Neural Netw Learn Syst https://doi.org/10.1109/TNNLS.2017.2709805
25. Zhang T, Zhou Z (2014) Large margin distribution machine. In: Proceedings of the 20th ACM International Conference on Knowledge Discovery and Data Mining, pp 313–322. https://doi.org/10.1145/2623330.2623710
26. Zhang T, Zhou Z (2020) Optimal margin distribution machine. IEEE Trans Knowledge Data Eng 32(6):1143–1156. https://doi.org/10.1109/TKDE.2019.2897662
27. Maurer A, Pontil M (2009) Empirical bernstein bounds and sample variance penalization. In: Proceedings of the 22nd Annual Conference on Learning Theory. Montreal, Canada, pp 1–9. https://arxiv.53yu.com/abs/0907.3740v1
28. Steinwart I, Hush D, Scovel C (2011) Training SVMs without offset. J Mach Learn Res 12(1):141–202. https://doi.org/10.5555/1953048.1953054
29. Vito ED, Rosasco L, Caponnetto A, Piana M, Verri A (2004) Some properties of regularized kernel methods. J Mach Learn Res 5:1363–1390
30. Schölkopf B, Herbrich R, Smola AJ (2001) A generalized representer theorem. In: International Conference on Computational Learning Theory, pp 416–426. https://doi.org/10.1007/3-540-44581-1_27
31. Steinwart I (2003) Sparseness of support vector machines. J Mach Learn Res. https://doi.org/10.1162/1532443041827925
32. Lee Y-J, Mangasarian OL (2001) RSVM: reduced support vector machines. In: Proceedings of the 2001 SIAM International Conference on Data Mining, pp 1–17. https://doi.org/10.1137/1.9781611972719.13
33. Keerthi SS, Chapelle O, DeCoste D (2006) Building support vector machines with reduced classifier complexity. J Mach Learn Res 7:1493–1515
34. Chapelle O (2007) Training a support vector machine in the primal. Neural comput 19(5):1155–1178. https://doi.org/10.1162/neco.2007.19.5.1155
35. Hoeffding W (1963) Probability inequalities for sums of bounded random variables. J Am Stat Assoc 58(301):13–30. https://doi.org/10.1080/01621459.1963.10500830
36. Golub GH, Heath M, Wahba G (1979) Generalized cross-validation as a method for choosing a good ridge parameter. Technometrics 21(2):215–223. https://doi.org/10.1080/00401706.1979.10489751
37. Kohavi R, *etal.* (1995) A study of cross-validation and bootstrap for accuracy estimation and model selection. In: International

Joint Conference on Artificial Intelligence, vol 14, pp 1137–1145. https://dl.acm.org/doi/10.5555/1643031.1643047

38. Fung GM, Mangasarian OL (2005) Proximal support vector machine classifiers. Mach Learn 59(1):77–97. https://doi.org/10.1007/s10994-005-0463-6
39. Zhou S, Cui J, Ye F, Liu H, Zhu Q (2013) New smoothing SVM algorithm with tight error bound and efficient reduced techniques. Comput Optimiz Appl 56(3):599–617. https://doi.org/10.1007/s10589-013-9571-6
40. Demiar J, Schuurmans D (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7(1):1–30. https://doi.org/10.5555/1248547.1248548

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.