Front. Math. China 2023, 18(6): 381–414
https://doi.org/10.3868/s140-DDD-023-0027-x

SURVEY ARTICLE

Analysis of loss functions in support vector machines

Huajun WANG, Naihua XIU

Department of Mathematics, School of Science, Beijing Jiaotong University, Beijing 100044, China

Abstract Support vector machines (SVMs) are a kind of important machine learning methods generated by the cross interaction of statistical theory and optimization, and have been extensively applied into text categorization, disease diagnosis, face detection and so on. The loss function is the core research content of SVM, and its variational properties play an important role in the analysis of optimality conditions, the design of optimization algorithms, the representation of support vectors and the research of dual problems. This paper summarizes and analyzes the 0-1 loss function and its eighteen popular surrogate loss functions in SVM, and gives three variational properties of these loss functions: subdifferential, proximal operator and Fenchel conjugate, where the nine proximal operators and fifteen Fenchel conjugates are given by this paper.

Keywords Support vector machines, loss function, subdifferential, proximal operator, Fenchel conjugate

MSC2020 68-02, 90-02, 68T05, 90C46

1 Introduction

Support vector machines (SVMs) were first proposed by Cortes and Vapnik [16] in 1995 and are widely used in text and image classification [11, 49, 53, 74], disease diagnosis [1, 12, 25, 35, 59, 65]. The basic idea is to find a hyperplane that separates the samples as correctly as possible while keeping the separated samples as far away from the hyperplane as possible. For the binary classification problem, given the training set $D = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^m$, where $\boldsymbol{x}_i \in \mathbb{R}^n$ is the input vector, $y_i \in \{-1, 1\}$ is the output label, and the goal of the SVM is to find the optimal hyperplane $\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b = 0$, given the training set, where

Translated from *Advances in Mathematics* (*China*), 2021, 50(6): 801–828
Corresponding author: Naihua XIU, E-mail: nhxiu@bjtu.edu.cn

$$\boldsymbol{w} \in \mathbb{R}^n, b \in \mathbb{R}.$$

For any new input vector $\boldsymbol{x}_{\text{new}}$, according to $\langle \boldsymbol{w}, \boldsymbol{x}_{\text{new}} \rangle + b > 0$, we predict that the corresponding label is $y_{\text{new}} = 1$. Otherwise, it is $y_{\text{new}} = -1$. In order to find the optimal hyperplane, the training data are considered in two cases: linearly divisible and linearly indivisible. For linearly separable training data, the unique optimal hyperplane is obtained by solving the following convex quadratic programming problem:

$$\min_{\boldsymbol{w}\in\mathbb{R}^n, b\in\mathbb{R}} \frac{1}{2}\|\boldsymbol{w}\|^2 \quad \text{s.t. } y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) \geqslant 1,\ i \in \mathbb{N}_m,$$

where $\mathbb{N}_m := \{1, 2, \ldots, m\}$. The above model is called hard interval SVM, because it requires that all training samples must be correctly separated. For linearly indistinguishable training data, the soft interval SVM optimization model is obtained by allowing some samples to not satisfy the constraints of the above model and minimizing the loss of those samples that do not satisfy the constraints in the objective function:

$$\min_{\boldsymbol{w}\in\mathbb{R}^n, b\in\mathbb{R}} \frac{1}{2}\|\boldsymbol{w}\|^2 + \lambda \sum_{i=1}^{m} \ell(1 - y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b)), \tag{1}$$

where $\lambda > 0$ is the penalty parameter, $\ell(t)$ is the loss function, $t := 1 - y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) \in \mathbb{R}$. The loss function is the core study of soft-interval SVM because it not only determines the sensitivity of the soft-interval SVM model to training data noise, but also affects the sparsity of the soft-interval SVM model. In literatures [4, 16, 22], Cortes and Vapnik et al. pointed out that the optimal soft-interval SVM optimization model is to minimize the number of erroneous samples of the training data, and the optimization model is

$$\min_{\boldsymbol{w}\in\mathbb{R}^n, b\in\mathbb{R}} \frac{1}{2}\|\boldsymbol{w}\|^2 + \lambda \sum_{i=1}^{m} \ell_{0/1}(1 - y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b)). \tag{2}$$

The mathematical expression for the 0-1 loss function is

$$\ell_{0/1}(t) = \begin{cases} 1, & t > 0; \\ 0, & t \leqslant 0. \end{cases}$$

It is a non-convex discontinuous bounded function, discontinuous at $t = 0$. However, the nonconvex discontinuous 0-1 loss function is included in the objective function of model (2), which makes traditional optimization theories and algorithms incapable of handling such problems. Therefore, during the two decades of soft-interval SVM development, it has been a hot topic of research in soft-interval SVM to construct proxy loss functions (convex or nonconvex loss functions) with better computational properties than the 0-1 loss function, taking into account the structure and characteristics of the training data. In constructing the loss function, it is important to consider its statistical properties compared with the 0-1 loss function on the one hand, and its computability in optimization

on the other hand. Unlike literatures [51, 56, 63, 75], which summarize the statistical aspects of the loss function, this paper focuses on three variational properties of the loss function, namely, subdifferentiation, neighborhood operator and Fenchel conjugate, in terms of optimization. In order to find the optimal solution of the soft interval SVM model, scholars have obtained rich theoretical and algorithmic results by using the above three variational properties of the loss function.

• Theories and algorithms based on subdifferentiation. The study of the theory and algorithm of the soft interval SVM model (1) by subdifferentiation of the loss function is mainly in three aspects. i) In the optimality theory, the Karush-Kuhn-Tucker (KKT) condition of the model (1) can be established by using subdifferentiation of the loss function to characterize the optimal solution, and the KKT condition can be used not only as a stopping condition for the algorithm, but also for designing efficient and feasible optimization algorithms [14, 16, 26, 57]. ii) In terms of representing the support vector, the non-zero nature of the subdifferential of the loss function is used to represent the support vector of the model (1) [14, 16, 26, 57]. Thus, the sparsity of model (1) is ensured and it is convenient to design fast and efficient algorithms because the optimal hyperplane obtained from the support vector is the same as the optimal hyperplane obtained from all training data [18]. iii) In terms of algorithm design, the subdifferentiation of the loss function enables the design of subgradient algorithms [10] and stochastic subgradient algorithms [50], etc. To facilitate the reader to understand the importance of subdifferentiation of the loss function in algorithm design, the hinge SVM subgradient algorithm [10] is given below. Given the kth iteration point $(\boldsymbol{w}^k; b^k)$, the iteration format is as follows.

$$\begin{aligned}\boldsymbol{w}^{k+1} &= \boldsymbol{w}^k + \gamma_k\big(\lambda A^\top \partial L_{\rm hl}(\boldsymbol{e} - A\boldsymbol{w}^k - b^k\boldsymbol{y}) + \boldsymbol{w}^k\big),\\ b^{k+1} &= b^k + \gamma_k\lambda \boldsymbol{y}^\top \partial L_{\rm hl}(\boldsymbol{e} - A\boldsymbol{w}^k - b^k\boldsymbol{y}),\end{aligned}$$

where γ_k denotes the iteration step, $L_{\rm hl}(\cdot) := \sum_{i=1}^m \ell_{\rm hl}(\cdot)$ denotes the hinge loss function, $\partial L_{\rm hl}(\cdot)$ denotes the subgradient of $L_{\rm hl}(\cdot)$.

• Theories and algorithms based on the neighborhood operator. The neighborhood operator of the loss function for the soft interval SVM model (1) is also studied in three aspects. i) In terms of optimality theory, for convex loss functions, the neighborhood stability condition established by the neighborhood operator of the loss function is the same as the KKT condition. For non-convex loss functions, the neighborhood stability condition is generally stronger than the KKT condition, which can also be used as a stopping condition and algorithm design [61, 62]. ii) In terms of representing support vectors, for convex loss functions, the support vector of the model (1) represented by the neighborhood point operator of the loss function is equivalent to the support vector represented by the subdifferential of the loss function. For non-convex loss functions, the support vectors expressed using the neighborhood point operator are generally represented as a subset of the support vectors using subdifferentiation [61]. iii) In terms of

algorithm design, for convex loss functions, the neighborhood point operator of the loss function enables the design of the semi-smooth Newton extended Lagrangian method [67], and for non-convex loss functions, the alternating direction multiplier method (ADMM) [62, 72], and so on. In order to facilitate the reader to understand the importance of the neighborhood point operator of the loss function in the algorithm design, the algorithm based on the SVM (0.2) neighborhood stabilization point design is given below. Literature [62] calls $(\boldsymbol{w}^*; b^*)$ the neighborhood stabilizer of SVM (0.2) if there exist vectors $\boldsymbol{\beta}^*, \boldsymbol{u}^* \in \mathbb{R}^m$ and parameters $\gamma > 0$ such that they satisfy the following expressions:

$$\begin{cases} \boldsymbol{w}^* + A^\top \boldsymbol{\beta}^* = \mathbf{0}, \\ \boldsymbol{y}^\top \boldsymbol{\beta}^* = 0, \\ \boldsymbol{u}^* + A\boldsymbol{w}^* + b^* \boldsymbol{y} = \boldsymbol{e}, \\ \mathrm{prox}_{\gamma\lambda L}(\boldsymbol{u}^* - \gamma\boldsymbol{\beta}^*) \ni \boldsymbol{u}^*. \end{cases}$$

Given the kth iteration point $(\boldsymbol{w}^k; b^k; \boldsymbol{u}^k; \boldsymbol{\beta}^k)$, the format of the neighborhood point algorithm iteration is as follows:

$$\begin{cases} \boldsymbol{w}^{k+1} = \boldsymbol{w}^k - \gamma(\boldsymbol{w}^k + A^\top \boldsymbol{\beta}^k), \\ \boldsymbol{b}^{k+1} = b^k - \gamma(\boldsymbol{y}^\top \boldsymbol{\beta}^k), \\ \boldsymbol{\beta}^{k+1} = \boldsymbol{\beta}^k - \gamma(\boldsymbol{u}^k + A\boldsymbol{w}^k + b^k \boldsymbol{y} - \boldsymbol{e}), \\ \boldsymbol{u}^{k+1} = \mathrm{prox}_{\gamma\lambda L}(\boldsymbol{u}^k - \gamma\boldsymbol{\beta}^k), \end{cases}$$

where $\mathrm{prox}_{\gamma\lambda L}(\cdot)$ is the neighborhood point operator of the 0-1 loss function $\ell_{0/1}(\cdot)$, whose specific expression is given in literature [62].

• Theories and algorithms based on the Fenchel conjugate. The Fenchel conjugate of the loss function is used to derive the pairwise model from the original model (1), therefore, the Fenchel conjugate of the loss function determines the analytic expression of the pairwise model. The solution of the dual model has two advantages. i) When the dual model is easier to solve than the original model, the optimal solution of the original problem can be obtained by designing a fast algorithm to solve the dual model. Since the pairwise gap between the original model and the dual model of the convex loss function is zero, these algorithms are usually suitable for convex loss functions. Typical algorithms include the pairwise coordinate descent method [20, 70] and the sequence minimization algorithm [8, 32–34, 41], etc. ii) By introducing the kernel function in the pairwise model, it can be further extended to deal with the nonlinear classification problem [38].

In summary, a good understanding of the variational properties of the existing soft-interval SVM loss functions can not only enable us to choose the appropriate loss functions when building soft-interval SVM models and designing algorithms, but also provide research ideas for constructing new loss functions. Therefore, it is necessary to comprehensively organize and comprehensively review the variational properties of the soft interval SVM loss function. In this paper, the 0-1 loss

function and its 18 commonly used SVM proxy loss functions are summarized and reviewed, with emphasis on their subdifferentiation, neighborhood operator and Fenchel conjugate, of which 9 neighborhood operators and 15 Fenchel conjugates are given in this paper.

2 SVM proxy loss function

This section introduces 18 commonly used soft-interval SVM proxy loss functions. For the convenience of the later discussion, we classify the proxy loss functions into four categories according to their convexity and smoothness, as shown in Fig. 1.

2.1 Convex non-smooth loss function

(1) The hinge loss function. In 1995, Cortes and Vapnik [16] used the hinge loss function when they built the first soft interval SVM model, which has the mathematical expression

$$\ell_{\mathrm{hl}}(t) = \begin{cases} t, & t > 0; \\ 0, & t \leqslant 0. \end{cases}$$

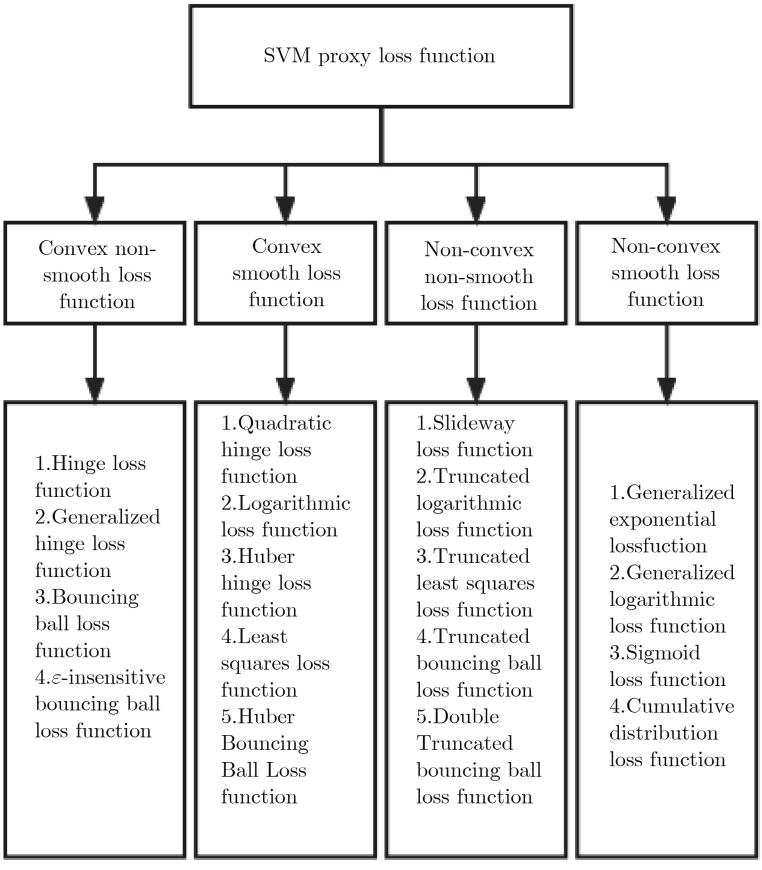

Fig. 1 SVM agent loss function classification

It is the best convex approximation of the 0-1 loss function [75], and is one of the most popular convex loss functions in soft interval SVM. Its function image is shown in Fig. 2(a). For $t \leqslant 0$ samples, its loss value is 0, which means it does not penalize the samples that are sufficiently correctly classified, and therefore has good sparsity. For samples with $t > 0$, the loss value is t, which means that outliers contribute a large weight to the corresponding SVM objective function, thus affecting the optimal hyperplane found, and thus are more sensitive to outliers [5, 7, 40, 52].

(2) Generalized hinge loss (GHL) function. In 2008, Bartlett and Wegkamp [2] generalized the hinge loss function and proposed the generalized hinge loss function, which has the mathematical expression

$$\ell_{\mathrm{gh}}(t) = \begin{cases} 1 + \eta(t-1), & t > 1; \\ t, & t \in (0, 1]; \\ 0, & t \leqslant 0, \end{cases}$$

where $\eta \geqslant 1$. The function image is shown in Fig. 2(a). When $\eta = 1$, the generalized hinge loss function degenerates to the hinge loss function. When $\eta > 1$, unlike the hinge loss function, the loss value is $1 + \eta(t-1)$ for samples with $t > 1$. Therefore, it has sparsity but is sensitive to outliers.

(3) Bouncing ball loss function. In 2013, Jumutc et al. [30] introduced the pinball loss function into the soft interval SVM, which has the mathematical expression

$$\ell_{\mathrm{pl}}(t) = \begin{cases} t, & t > 0; \\ -\tau t, & t \leqslant 0, \end{cases}$$

where $\tau \in [0, 1]$. The image of the function is shown in Fig. 2(b). When $\tau = 0$, the bouncing ball loss function degenerates to the hinge loss function. When $\tau \in (0, 1]$, unlike the hinge loss function, the loss value is $-\tau t$ for $t \leqslant 0$ samples. Therefore, it is not sparse and sensitive to outliers [26, 28, 29, 58].

(4) ε-insensitive pinball loss function. To overcome the drawback that the pinball loss function does not have sparsity, in 2014, Huang et al. [26] proposed the ε-insensitive pinball loss function, which has the mathematical expression

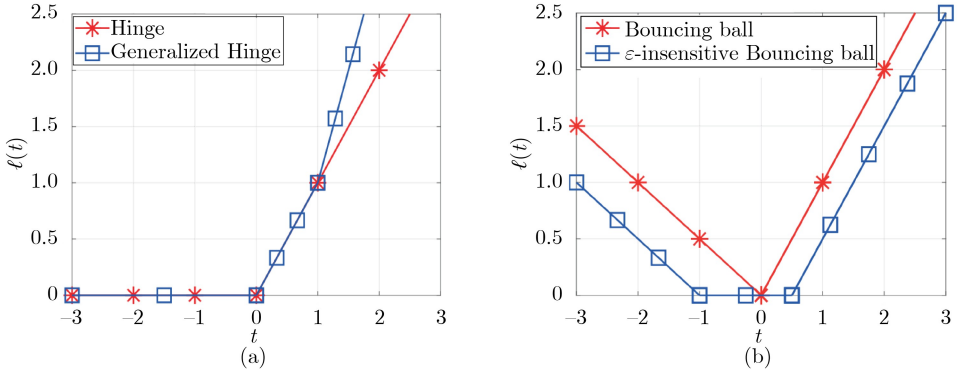

Fig. 2 Convex non-smooth loss function schematic

$$\ell_{\mathrm{ip}}(t)=\begin{cases}t-\varepsilon, & t>\varepsilon;\\ 0, & t\in\left[-\dfrac{\varepsilon}{\tau},\varepsilon\right];\\ -\tau\left(t+\dfrac{\varepsilon}{\tau}\right), & t<-\dfrac{\varepsilon}{\tau},\end{cases}$$

where $\tau\in[0,1]$, $\varepsilon>0$. The image of the function is shown in Fig. 2(b). Unlike the bouncing ball loss function, for samples with $t<-\frac{\varepsilon}{\tau}$, the loss value is $-\tau(t+\frac{\varepsilon}{\tau})$. The samples with $t\in[-\frac{\varepsilon}{\tau},\varepsilon]$ are not penalized. For samples with $t>\varepsilon$, the loss value is $t-\varepsilon$. Thus, it is sparse but sensitive to outliers.

2.2 Convex smooth loss function

(5) Quadratic hinge loss (squared hinge loss) function. To overcome the drawback that the hinge loss function is non-smooth at $t=0$, in 1995, Cortes and Vapnik [16] proposed the quadratic hinge loss function, which has the mathematical expression

$$\ell_{\mathrm{sh}}(t)=\begin{cases}t^2, & t>0;\\ 0, & t\leqslant 0.\end{cases}$$

The function image is shown in Fig. 3(a). Unlike the hinge loss function, for samples with $t>0$, the loss value is t^2. Therefore, it is sparse but sensitive to outliers [9, 10, 31, 36, 37, 55, 71, 73].

(6) Huber hinge loss function. To achieve the smoothness of the hinge loss function at $t=0$, in 2007, Chapelle [10] proposed the Huber hinge loss function, which has the mathematical expression

$$\ell_{\mathrm{hh}}(t)=\begin{cases}t-\dfrac{\delta}{2}, & t>\delta;\\ \dfrac{t^2}{2\delta}, & t\in[0,\delta];\\ 0, & t<0,\end{cases}$$

where $\delta>0$. The image of the function is shown in Fig. 3(a). Unlike the hinge

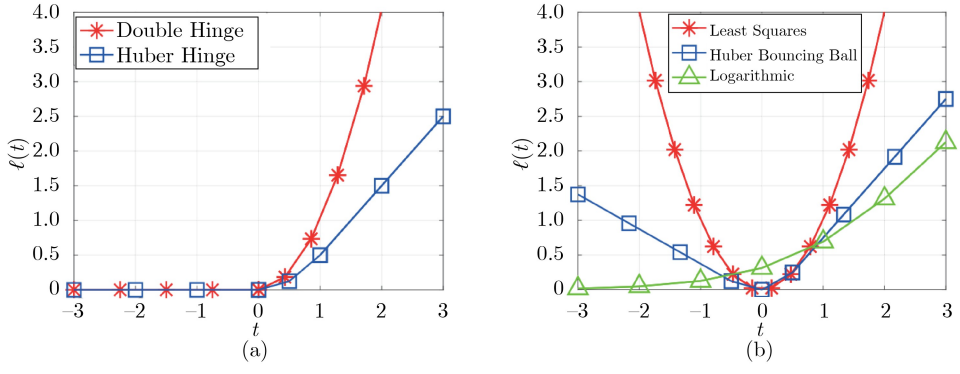

Fig. 3 Convex smooth loss function schematic

loss function, for samples with $t \in [0, \delta]$, it achieves smoothness at $t = 0$ through the loss value $\frac{t^2}{2\delta}$. For samples with $t > \delta$, the loss value is $t - \frac{\delta}{2}$. Thus, it is sparse but sensitive to outliers [39, 66, 69].

(7) Logarithmic (Logistic) loss function. In 1998, Wahba [60] introduced the logistic loss function into the soft interval SVM, which has the mathematical expression

$$\ell_{\rm ll}(t) = \log(1 + \exp(t - 1)), \quad t \in \mathbb{R}.$$

The image of the function is shown in Fig. 3(b). Unlike the hinge loss function, it imposes a logarithmic penalty on all samples. Therefore, it is not sparse and sensitive to outliers.

(8) Least squares loss (LSL) function. In 1999, Suykens and Vandewalle [57] introduced the least squares loss function into the soft interval SVM, which has the mathematical expression

$$\ell_{\rm ls}(t) = t^2, \quad t \in \mathbb{R}.$$

The image of the function is shown in Fig. 3(b). Unlike the bouncing ball loss function, it imposes a quadratic penalty on all samples. Therefore, it is not sparse and sensitive to outliers [17, 21, 33, 43, 76, 77].

(9) Huber pinball loss function. In order to overcome the drawback of non-smoothness of the pinball loss function at $t = 0$, in 2020, Zhu et al. [78] proposed the Huber pinball loss function, which has the mathematical expression

$$\ell_{\rm hp}(t) = \begin{cases} t - \dfrac{\delta}{2}, & t > \delta; \\ \dfrac{t^2}{2\delta}, & t \in (0, \delta]; \\ \dfrac{\tau t^2}{2\delta}, & t \in [-\delta, 0]; \\ -\tau\left(t + \dfrac{\delta}{2}\right), & t < -\delta, \end{cases}$$

where $\tau \in [0, 1]$, $\delta > 0$. The image of the function is shown in Fig. 3(b). Unlike the bouncing ball loss function, the Huber bouncing ball loss function achieves smoothness at $t = 0$ by applying a quadratic penalty to the samples of $[-\delta, \delta]$. For samples with $t < -\delta$, the loss value is $-\tau(t + \frac{\delta}{2})$. For $t > -\delta$, the loss value is $t - \frac{\delta}{2}$. Therefore, it is not sparse and sensitive to outliers.

Since the above nine loss functions are convex, their corresponding soft interval SVM models are easy to solve [45]. However, the convex loss functions are usually unbounded, which makes them sensitive to outliers in the training data. To overcome this drawback, scholars have obtained the non-convex loss function described below by placing an upper bound on the loss, i.e., forcing the loss to

stop increasing after a certain point.

2.3 Non-convex non-smooth loss function

(10) Slideway loss (ramp loss) function. To overcome the drawback that the hinge loss function is sensitive to outliers, in 2003, Shen et al. [54] proposed the ramp loss function, which has the mathematical expression

$$\ell_{\mathrm{rl}}(t) = \begin{cases} 1, & t > 1; \\ t, & t \in (0, 1]; \\ 0, & t \leqslant 0. \end{cases}$$

The chute loss function is one of the most popular nonconvex loss functions in soft interval SVM. Its function image is shown in Fig. 4(a). Unlike the hinge loss function, the loss value is 1 for samples with $t > 1$ and these samples are non-support vectors [54]. Therefore, it has better sparsity than the hinge loss function and is robust to outliers [6, 19, 44, 74]. In addition, the literature [15, 27, 64] investigates the slideway loss function with adjustable parameters $\mu > 0$.

(11) Truncated logistic loss function. To overcome the shortcomings of the logistic loss function which is sensitive to outliers, in 2011, Park and Liu [46] proposed the truncated logistic loss function, which has the mathematical expression

$$\ell_{\mathrm{tll}}(t) = \begin{cases} \log(1 + \exp(-\nu)), & t > 1 - \nu; \\ \log(1 + \exp(t - 1)), & t \leqslant 1 - \nu, \end{cases}$$

where $\nu < 1$. Its function image is shown in Fig. 4(a). Unlike the logarithmic loss function, its loss value is $\log(1 + \exp(-\nu))$ for samples with $t > 1 - \nu$, and these samples are non-support vectors [46]. Therefore, they are sparse and robust to outliers.

(12) Truncated least square loss (TLSL) function. To overcome the drawback that the least-squares loss function is not sparse and sensitive to outliers, in 2016, Liu et al. [42] proposed the truncated least-squares loss function, which has the mathematical expression

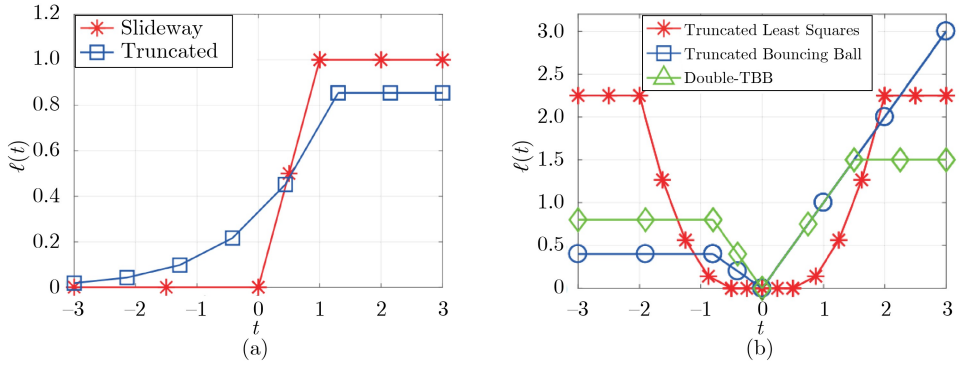

Fig. 4 Non-convex non-smooth loss function schematic

$$\ell_{\mathrm{tll}}(t)=\begin{cases}(\mu-\varepsilon)^2, & |t|>\mu;\\ (|t|-\varepsilon)^2, & |t|\in[\varepsilon,\mu];\\ 0, & |t|\in[0,\varepsilon),\end{cases}$$

where $0<\varepsilon<\mu$. The function image is shown in Fig. 4(b). Unlike the least-squares loss function, it does not penalize samples with $|t|\in[0,\varepsilon)$. For samples with $|t|\in[\varepsilon,\mu]$, the loss value is $(|t|-\varepsilon)^2$. For samples with $|t|>\mu$, the loss value is $(\mu-\varepsilon)^2$, and these samples are non-support vectors [42]. Therefore, they are sparse and robust to outliers.

(13) Truncated bouncing ball loss function. To overcome the drawback that the pinball loss function is not sparse, in 2017, Shen et al. proposed the truncated pinball loss function, which has the mathematical expression

$$\ell_{\mathrm{tp}}(t)=\begin{cases}t, & t>0;\\ -\tau t, & t\in[-\kappa,0];\\ \tau\kappa, & t<-\kappa,\end{cases}$$

where $\tau\in[0,1]$, $\kappa>0$. The image of the function is shown in Fig. 4(b). Unlike the bouncing ball loss function, for samples with $t<-\kappa$, the loss value is constant $\tau\kappa$, and these samples are non-support vectors [55]. Thus are sparse but sensitive to outliers.

(14) Double-truncated pinball loss function. In order to improve the robustness of the truncated pinball loss function to outliers, in 2018, Yang and Dong [68] proposed the bi-truncated pinball loss function, which has the mathematical expression

$$\ell_{\mathrm{btp}}(t)=\begin{cases}\mu, & t>\mu;\\ t, & t\in(0,\mu);\\ -\tau t, & t\in(-\kappa,0];\\ \tau\kappa, & t\leqslant-\kappa,\end{cases}$$

where $\tau\in[0,1]$, $\kappa,\mu>0$. The image of the function is shown in Fig. 4(b). Unlike the truncated bouncing ball loss function, the loss value is μ for samples with $t>\mu$, and these samples are non-support vectors [68]. Therefore, it is sparse and robust to outliers.

2.4 Non-convex smooth loss function

(15) Generalized exponential loss (GEL) function. In order to overcome the drawback that the slideway loss function is non-smooth at $t=0,1$, in 2016, Feng et al. [22] proposed the generalized exponential loss function, which has the mathematical expression

$$\ell_{\mathrm{gel}}(t)=\begin{cases}\sigma^2\left(1-\exp\left(-\left(\dfrac{t}{\sigma}\right)^2\right)\right), & t>0;\\ 0, & t\leqslant 0,\end{cases}$$

where $\sigma>0$. The function image is shown in Fig. 5(a). Unlike the sliding loss

function, for $t \geqslant 0$ samples, the loss value is $\sigma^2(1-\exp(-(\frac{t}{\sigma})^2))$ and as t increases $\ell_{\text{gel}}(t)$ converges to σ^2. Thus, there is sparsity and the parameter σ controls the robustness to outliers.

(16) Generalized logistic loss (GLL) function. In 2016, another loss function proposed by Feng et al. [22] is the generalized logistic loss function, which has the mathematical expression

$$\ell_{\text{gll}}(t) = \begin{cases} \sigma^2 \log\left(1+\left(\dfrac{t}{\sigma}\right)^2\right), & t > 0; \\ 0, & t \leqslant 0, \end{cases}$$

where $\sigma > 0$. The function image is shown in Fig. 5(a). Unlike the sliding loss function, for $t > 0$ samples, the loss value is $\sigma^2 \log(1+(\frac{t}{\sigma})^2)$ and as t increases $\ell_{\text{gll}}(t)$ tends to be σ^2. Therefore, it is sparse and the parameter σ controls the robustness to outliers.

(17) Sigmoid loss function. In 2003, Pérez-Cruz and Navia-Vázquez [47] introduced the Sigmoid loss function into the soft interval SVM, which has the mathematical expression

$$\ell_{\text{sl}}(t) = \frac{1}{1+\exp(-\beta t)}, \quad t \in \mathbb{R},$$

where $\beta > 0$. The image of the function is shown in Fig. 5(b). Unlike the slide-loss function, it penalizes all samples. For $t \geqslant 0$ samples, the upper bound of the loss value is 1. Thus it is not sparse but robust to outliers.

(18) Cumulative distribution loss (CDL) function. In 2019, Ghanbari et al. [24] proposed the cumulative distribution loss function to smooth the approximate 0-1 loss function, which has the mathematical expression

$$\ell_{\text{cd}}(t) = \int_{-\infty}^{t} \frac{1}{2\pi} \exp\left(-\frac{\gamma^2}{2}\right) \mathrm{d}\gamma, \quad t \in \mathbb{R}.$$

The image of the function is shown in Fig. 5(b). Unlike the slide-loss function, it penalizes all samples and the upper bound of the loss value is 1 as t increases. Therefore, it is not sparse but robust to outliers.

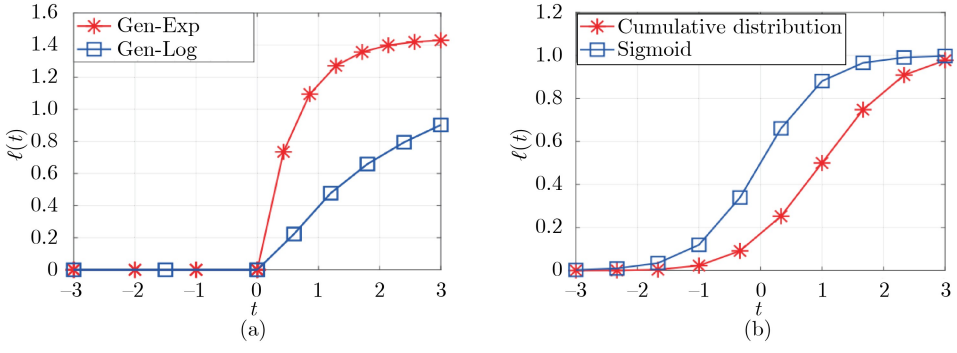

Fig. 5 Non-convex smooth loss function schematic

3 Variational properties of the proxy loss function

In the previous section we give 18 commonly used loss functions for SVM agents. In this section, we introduce and give three variational properties of these loss functions, namely subdifferential, neighborhood point operator and Fenchel conjugate, which play an important role in model (0.1) optimality condition inscription, optimization algorithm design, support vector representation and pairwise problem study.

3.1 Subdifferential

In this subsection, we introduce the subdifferentiation of the loss function of the above 18 SVM agents. If the loss function is a differentiable function, then it has a gradient at any point. If the loss function is not differentiable at a point, then the gradient at that point does not exist, so the concept of subdifferentiation needs to be introduced.

Definition 3.1 [13, 48] Given a function $f:\mathbb{R}\to\overline{\mathbb{R}}$ and a point $u\in\mathbb{R}$, if $f(u)$ is finite, we call

(i) $\boldsymbol{v}\in\mathbb{R}$ is a regular subdifferential of f at u if

$$\widehat{\partial}f(u):=\left\{\boldsymbol{v}\in\mathbb{R}:\liminf_{z\to u,z\neq u}\frac{f(z)-f(u)-\langle\boldsymbol{v},z-u\rangle}{\|z-u\|}\geqslant 0.\right.$$

(ii) $\boldsymbol{v}\in\mathbb{R}$ is the limiting subdifferential of f at u, if

$$\partial f(u):=\limsup_{z\overset{f}{\to}u}\widehat{\partial}f(z)=\left\{\boldsymbol{v}\in\mathbb{R}:\ \exists z_j\overset{f}{\to}u,\ \mathrm{v}_j\in\widehat{\partial}f(z_j)\text{ and }\mathrm{v}_j\to\mathrm{v}\right\}.$$

(iii) $\boldsymbol{v}\in\mathbb{R}$ is the Clarke subdifferential of f at u, if

$$\partial^C f(u):=\{\boldsymbol{v}\in\mathbb{R}:\ \boldsymbol{v}\xi\leqslant f^\circ(u;\xi),\ \forall\xi\ \in\mathbb{R}\},$$

where $f^\circ(u;\xi)$ denotes the Clarke directional derivative of f at u along the direction ξ, i.e.,

$$f^\circ(u;\xi):=\limsup_{z\to u,\rho\downarrow 0}\frac{f(z+\rho\xi)-f(z)}{\rho}.$$

When f is a convex function, the limit subdifferential degenerates to the subdifferential of the convex function, i.e.,

$$\partial f(u)=\{\boldsymbol{v}\in\mathbb{R}:\ f(z)-f(u)\geqslant\langle\boldsymbol{v},z-u\rangle,\ \ \forall z\in\mathbb{R}\}.$$

When f is non-convex, the Clarke subdifferential is the closed convex package [48] of the limiting subdifferential.

3.1.1 Subdifferentiation of convex nonsmooth loss function

(1) (Hinge loss function) In 1995, Cortes and Vapnik [16] gave the subdifferentiation

of the hinge loss function

$$\partial \ell_{\mathrm{hl}}(t) = \begin{cases} 1, & t > 0; \\ [0,1], & t = 0; \\ 0, & t < 0. \end{cases}$$

The next differential image is shown in Fig. 6(a). When $t = 0$, the second derivative belongs to the closed interval $[0,1]$; when $t > 0$, its gradient is 1; when $t < 0$, its gradient is 0.

(2) (Generalized hinge loss function) In 2008, Bartlett et al. [2] gave the subdifferentiation of the generalized hinge loss function

$$\partial \ell_{\mathrm{gh}}(t) = \begin{cases} \eta, & t > 1; \\ [1,\eta], & t = 1; \\ 1, & t \in (0,1); \\ [0,1], & t = 0; \\ 0, & t < 0, \end{cases}$$

where $\eta \geqslant 1$. The next differential image is shown in Fig. 6(a). When $\eta > 1$, unlike the subdifferential of the hinge loss function, when $t = 1$, the subdifferential belongs to the closed interval $[1,\eta]$; when $t > 1$, its gradient is η.

(3) (Bouncing ball loss function) In 2013, Jumutc et al. [30] gave the subdifferential of the Bouncing ball loss function

$$\partial \ell_{\mathrm{pl}}(t) = \begin{cases} 1, & t > 0; \\ [-\tau,1], & t = 0; \\ -\tau, & t < 0, \end{cases}$$

where $\tau \in [0,1]$. The second differential image is shown in Fig. 6(b). Unlike the subdifferential of the hinge loss function, when $t = 0$, the subdifferential belongs to the closed interval $[-\tau,1]$; when $t < 0$, its gradient is $-\tau$.

(4) (ε-insensitive bouncing ball loss function) In 2014, Huang et al. [26] give the subdifferential of ε-insensitive bouncing ball loss function

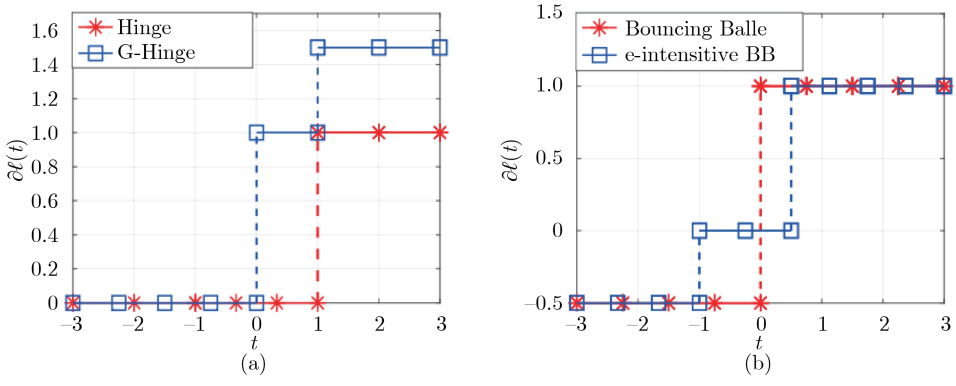

Fig. 6 Subdifferential diagram of convex nonsmooth loss function where the set of subdifferentials at the non-integrable points is shown by dashed lines

$$\partial \ell_{\mathrm{ip}}(t)=\begin{cases}1, & t>\varepsilon;\\ [0,1], & t=\varepsilon;\\ 0, & t\in\left(-\dfrac{\varepsilon}{\tau},\varepsilon\right);\\ [-\tau,0], & t=-\dfrac{\varepsilon}{\tau};\\ -\tau, & t<-\dfrac{\varepsilon}{\tau},\end{cases}$$

where $\tau\in[0,1]$, $\varepsilon>0$. The second differential image is shown in Fig. 6(b). Unlike the subdifferential of the bouncing ball loss function, when $t=\varepsilon$, the subdifferential belongs to the closed interval $[0,1]$; when $t\in(-\frac{\varepsilon}{\tau},\varepsilon)$, its gradient is 0; when $t=-\frac{\varepsilon}{\tau}$, the second derivative belongs to the closed interval $[-\tau,0]$.

3.1.2 Gradient of convex smooth loss function

(5) (Double hinge loss function) In 1995, Cortes and Vapnik [16] gave the gradient of the Double hinge loss function

$$\nabla \ell_{\mathrm{sh}}(t)=\begin{cases}2t, & t\geqslant 0;\\ 0, & t<0.\end{cases}$$

Its gradient image is shown in Fig. 7(a). Unlike the subdifferentiation of the hinge loss function, when $t\geqslant 0$, its gradient is $2t$.

(6) (Huber hinge loss function) 2007, Chapelle [10] gave the gradient of the Huber hinge function

$$\nabla \ell_{\mathrm{hh}}(t)=\begin{cases}1, & t>\delta;\\ \dfrac{t}{\delta}, & t\in[0,\delta];\\ 0, & t<0,\end{cases}$$

where $\delta>0$. Its gradient image is shown in Fig. 7(a). Unlike the subdifferentiation of the hinge loss function. When $t\in[0,\delta]$, its gradient is $\frac{t}{\delta}$.

(7) (Logarithmic loss function) In 1998, Wahba [60] gave the gradient of the logarithmic loss function

$$\nabla \ell_{\mathrm{ll}}(t)=1-\frac{1}{1+\exp(t-1)},\quad t\in\mathbb{R}.$$

Its gradient image is shown in Fig. 7(b). Unlike the subdifferential of the hinge loss function, its gradient is $1-\frac{1}{1+\exp(t-1)}$.

(8) (Least squares loss function) In 1999, Suykens and Vandewalle [57] gave the gradient of the least squares loss function

$$\nabla \ell_{\mathrm{ls}}(t)=2t,\quad t\in\mathbb{R}.$$

Its gradient image is shown in Fig. 7(b). Unlike the subdifferential of the bouncing ball loss function, its gradient is $2t$.

(9) (Huber bouncing ball loss function) In 2020, Zhu et al. [78] gave the gradient of the Huber bouncing ball loss function

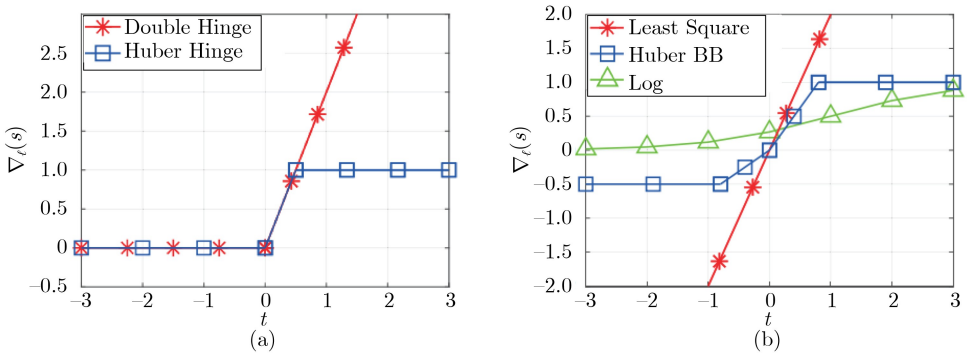

Fig. 7 Gradient diagram of convex smooth loss function

$$\nabla \ell_{\mathrm{hp}}(t) = \begin{cases} 1, & t > \delta; \\ \dfrac{t}{\delta}, & t \in (0, \delta]; \\ \dfrac{\tau t}{\delta}, & t \in (-\delta, 0]; \\ -\tau, & t \leqslant -\delta, \end{cases}$$

where $\tau \in [0, 1]$, $\delta > 0$. Its gradient image is shown in Fig. 7(b). Unlike the subdifferential of the bouncing ball loss function, when $t \in (0, \delta]$, its gradient is $\frac{t}{\delta}$; when $t \in (-\delta, 0]$, its gradient is $\frac{\tau t}{\delta}$.

3.1.3 Clarke subdifferential of nonconvex nonsmooth loss function

(10) (Slideway loss function) In 2003, Shen et al. [54] gave the Clarke subdifferential of the slideway loss function

$$\partial^C \ell_{\mathrm{rl}}(t) = \begin{cases} 0, & t > 1; \\ [0, 1], & t = 1; \\ 1, & t \in (0, 1); \\ [0, 1], & t = 0; \\ 0, & t < 0. \end{cases}$$

Its Clarke subdifferential image is shown in Fig. 8(a). Unlike the subdifferential of the hinge loss function, when $t = 1$, its Clarke subdifferential belongs to the closed interval $[0, 1]$; when $t > 1$, its gradient is 0.

(11) (Truncated logarithmic loss function) In 2011, Park and Liu [46] gave the truncated logarithmic loss function Clarke subdifferential

$$\partial^C \ell_{\mathrm{tll}}(t) = \begin{cases} 0, & t > 1 - \nu; \\ \left[0, 1 - \dfrac{1}{1 + \exp(-\nu)}\right], & t = 1 - \nu; \\ 1 - \dfrac{1}{1 + \exp(t - 1)}, & t < 1 - \nu, \end{cases}$$

where $\nu < 1$. Its Clarke subdifferential image is shown in Fig. 8(a). Unlike the gradient of the logarithmic loss function, when $t > 1 - \nu$, its gradient is 0; when $t = 1 - \nu$, its Clarke subdifferential belongs to the closed interval

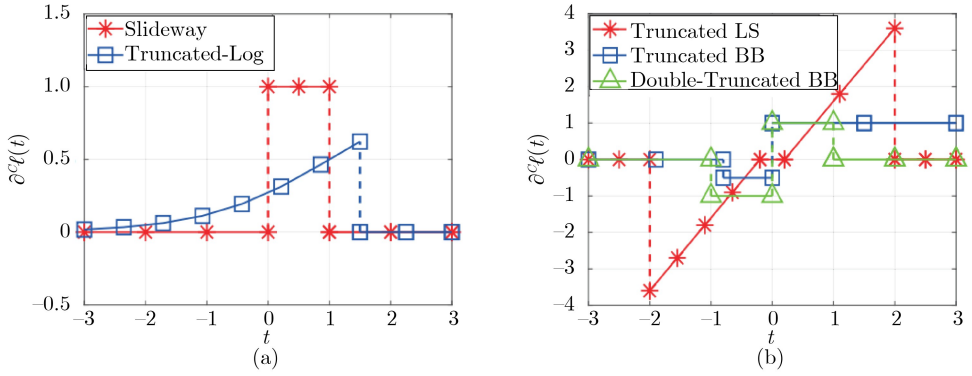

Fig. 8 Non-convex non-smooth loss function of Clarke subdifferential schematic where the set of Clarke subdifferentials at the non-differentiable points is shown by dashed lines

$[0, 1 - \frac{1}{1+\exp(-\nu)}]$.

(12) (Truncated least-squares loss function) In 2016, Liu et al. [42] gave the Clarke subdifferential of the truncated least-squares loss function

$$\partial^C \ell_{\text{tll}}(t) = \begin{cases} 0, & |t| > \mu; \\ 2\text{sgn}(t)(|t| - \varepsilon), & |t| \in [\varepsilon, \mu); \\ 0, & |t| \in [0, \varepsilon); \\ [0, 2(\mu - \varepsilon)], & t = \mu; \\ [2(-\mu + \varepsilon), 0], & t = -\mu, \end{cases}$$

where $0 < \varepsilon < \mu$, $\text{sgn}(\cdot)$ is the sign function. Its Clarke subdifferential image is shown in Fig. 8(b). Unlike the gradient of the least-squares loss function, when $|t| > \mu$, its gradient is 0; when $|t| \in [\varepsilon, \mu)$, its gradient is $2\text{sgn}(t)(|t| - \varepsilon)$; when $|t| \in [0, \varepsilon)$, its gradient is 0; when $t = \mu$, its Clarke subdifferential belongs to the closed interval $[0, 2(\mu - \varepsilon)]$; when $t = -\mu$, its Clarke subdifferential belongs to the closed interval $[2(-\mu + \varepsilon), 0]$.

(13) (Truncated bouncing ball loss function) In 2017, Shen et al. [55] gave the Clarke subdifferential of the truncated bouncing ball loss function

$$\partial^C \ell_{\text{tp}}(t) = \begin{cases} 1, & t > 0; \\ [-\tau, 1], & t = 0; \\ -\tau, & t \in (-\kappa, 0); \\ [-\tau, 0], & t = -\kappa; \\ 0, & t < -\kappa, \end{cases}$$

where $\tau \in [0, 1]$, $\kappa > 0$. Its Clarke subdifferential image is shown in Fig. 8(b). Unlike the subdifferential of the bouncing ball loss function, when $t = -\kappa$, its Clarke subdifferential belongs to the closed interval $[-\tau, 0]$; when $t < -\kappa$, its gradient is 0.

(14) (Double truncated bouncing ball loss function) In 2018, Yang et al. [68] gave the Clarke subdifferential of the double truncated bouncing ball loss func-

tion

$$\partial^C \ell_{\text{btp}}(t) = \begin{cases} 0, & t > \mu; \\ [0,1], & t = \mu; \\ 1, & t \in (0,\mu); \\ [-\tau,1], & t = 0; \\ -\tau, & t \in (-\kappa,0); \\ [-\tau,0], & t = -\kappa; \\ 0, & t < -\kappa, \end{cases}$$

where $\tau \in [0,1]$, $\kappa, \mu > 0$. Its Clarke subdifferential image is shown in Fig. 8(b). Unlike the Clarke subdifferential of the truncated bouncing ball loss function, when $t = \mu$, its Clarke subdifferential belongs to the closed interval $[0,1]$; when $t > \mu$, its gradient is 0.

3.1.4 Gradient of non-convex smooth loss function

(15) (Generalized exponential loss function) 2016, Feng et al. [22] gave the gradient of the generalized exponential loss function

$$\nabla \ell_{\text{gel}}(t) = \begin{cases} 2t \exp\left(-\left(\frac{t}{\sigma}\right)^2 \right), & t > 0; \\ 0, & t \leqslant 0, \end{cases}$$

where $\sigma > 0$. Its gradient image is shown in Fig. 9(a). Unlike the Clarke subdifferential of the slide loss function, the gradient is $2t\exp(-(\frac{t}{\sigma})^2)$ when $t > 0$.

(16) (Generalized logarithmic loss function) 2016, Feng et al. [22] gave the gradient of the generalized logarithmic loss function

$$\nabla \ell_{\text{gll}}(t) = \begin{cases} \dfrac{2t}{1 + (t/\sigma)^2}, & t > 0; \\ 0, & t \leqslant 0, \end{cases}$$

where $\sigma > 0$. Its gradient image is shown in Fig. 9(a). Unlike the Clarke subdifferential of the slide loss function, when $t > 0$, its gradient is $\frac{2t}{1+(t/\sigma)^2}$.

(17) (Sigmoid loss function) In 2003, Pérez-Cruz et al. [47] gave the gradient

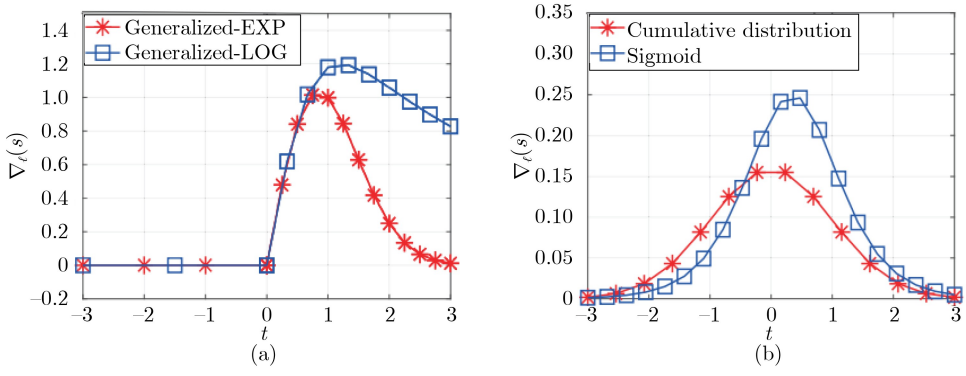

Fig. 9 Non-convex smooth loss function of the gradient schematic

of the Sigmoid loss function

$$\nabla \ell_{\mathrm{sl}}(t) = \frac{\beta \exp(-\beta t)}{(1+\exp(-\beta t))^2}, \quad t \in \mathbb{R},$$

where $\beta > 0$. Its gradient image is shown in Fig. 9(b). Unlike the Clarke subdifferential of the slide loss function, its gradient is $\frac{\beta \exp(-\beta t)}{(1+\exp(-\beta t))^2}$.

(18) (Cumulative distribution loss function) In 2019, Ghanbariti et al. [24] gave the gradient of the cumulative distribution loss function

$$\nabla \ell_{\mathrm{cd}}(t) = \frac{\exp(-t^2/2)}{2\pi}, \quad t \in \mathbb{R}.$$

Its gradient image is shown in Fig. 9(b). Unlike the Clarke subdifferential of the slide loss function, its gradient is $\frac{\exp(-t^2/2)}{2\pi}$.

3.2 Neighborhood point operator

Since 6 of the 18 SVM proxy loss functions introduced in Section 2 contain exponential or logarithmic loss functions and these 6 loss functions do not have explicit expressions for the neighborhood operator. Therefore, we first give the definition of the neighborhood operator, and then give the explicit expressions and graphs of the neighborhood operator for the other 12 loss functions.

Definition 3.2 [3] Let $f : \mathbb{R} \to \overline{\mathbb{R}}$ be a normal lower semicontinuous function, then the neighborhood point operator of f at a given $s \in \mathbb{R}$ with respect to the parameter $\alpha > 0$ is defined as

$$\mathrm{prox}_f(s) := \arg\min_{v \in \mathbb{R}} f(v) + \frac{1}{2\alpha}(v-s)^2.$$

When f is a convex function, its neighborhood operator is single-valued. When f is non-convex, its neighborhood operator may be multi-valued.

3.2.1 Neighborhood operators of convex nonsmooth loss functions

(1) (Hinge loss function) In 2020, Yan and Li [67] gave the neighborhood point operator for the hinge loss function

$$\mathrm{prox}_{\ell_{\mathrm{hl}}}(s) = \begin{cases} s-\alpha, & s > \alpha; \\ 0, & s \in [0, \alpha]; \\ s, & s < 0. \end{cases}$$

The image of its neighborhood point operator is shown in Fig. 10(a). When $s > \alpha$, its neighborhood operator is $s - \alpha$; when $s \in [0, \alpha]$, its neighborhood operator is 0; when $s < 0$, its neighborhood operator is s.

(2) (Generalized hinge loss function) In this paper, the authors proved to obtain the proximity point operator of the generalized hinge loss function

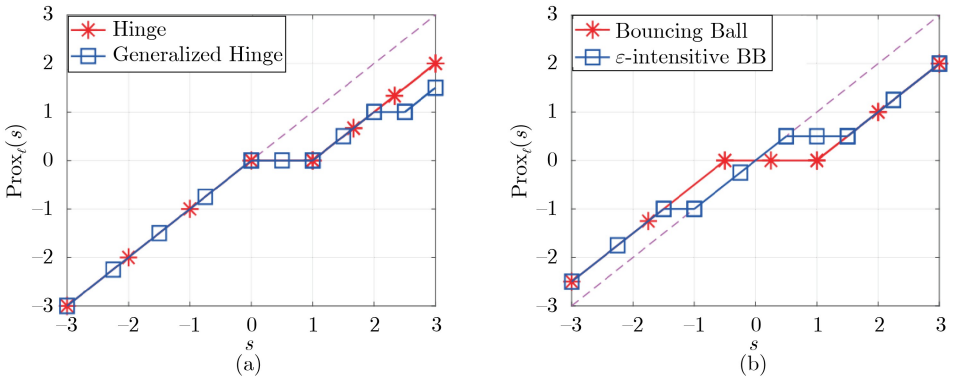

Fig. 10 Schematic diagram of the neighborhood point operator for convex nonsmooth loss function

$$\operatorname{prox}_{\ell_{\mathrm{gh}}}(s)=\begin{cases}s-\alpha\eta, & s>1+\alpha\eta;\\ 1, & s\in(1+\alpha,1+\alpha\eta];\\ s-\alpha, & s\in(\alpha,1+\alpha];\\ 0, & s\in(0,\alpha];\\ s, & s\leqslant 0,\end{cases}$$

where $\eta>1$. The image of its neighborhood point operator is shown in Fig. 10(a). Unlike the neighborhood point operator of the hinge loss function, when $s\in(1+\alpha,1+\alpha\eta]$, its neighborhood point operator is 1; when $s>1+\alpha\eta$, its neighborhood point operator is $s-\alpha\eta$.

(3) (Bouncing ball loss function) In this paper, the authors proved that the proximity operator of the bouncing ball loss function

$$\operatorname{prox}_{\ell_{\mathrm{pl}}}(s)=\begin{cases}s-\alpha, & s>\alpha;\\ 0, & s\in[-\tau\alpha,\alpha];\\ s+\tau\alpha, & s<-\tau\alpha,\end{cases}$$

where $\tau\in[0,1]$. Its neighborhood point operator is shown in Fig. 10(b). Unlike the proximity operator of the hinge loss function, when $s\in[-\tau\alpha,\alpha]$, its proximity operator is 0; when $s<-\tau\alpha$, its proximity operator is $s+\tau\alpha$.

(4) (ε-insensitive bouncing ball loss function) In this paper, the authors prove that the neighborhood point operator of ε-insensitive bouncing ball loss function

$$\operatorname{prox}_{\ell_{\mathrm{ip}}}(s)=\begin{cases}s-\alpha, & s>\alpha+\varepsilon;\\ \varepsilon, & s\in[\varepsilon,\alpha+\varepsilon];\\ s, & s\in\left[-\dfrac{\varepsilon}{\tau},\varepsilon\right);\\ -\dfrac{\varepsilon}{\tau}, & s\in\left[-\dfrac{\varepsilon}{\tau}-\tau\alpha,-\dfrac{\varepsilon}{\tau}\right);\\ s+\tau\alpha, & s<-\dfrac{\varepsilon}{\tau}-\tau\alpha,\end{cases}$$

where $\tau\in[0,1]$, $\varepsilon>0$. The image of its neighborhood point operator is shown in Fig. 10(b). Unlike the neighborhood operator of the bouncing ball loss func-

tion, when $s \in [\varepsilon, \alpha+\varepsilon]$, its neighborhood operator is ε; when $s \in [-\frac{\varepsilon}{\tau}, \varepsilon)$, its neighborhood operator is s; when $s \in [-\frac{\varepsilon}{\tau}, \varepsilon)$, the operator is s; when $s \in [-\frac{\varepsilon}{\tau}-\tau\alpha, -\frac{\varepsilon}{\tau})$, the operator is $-\frac{\varepsilon}{\tau}$.

3.2.2 Neighborhood point operator for convex smooth loss function

(5) (Double hinge loss function) In this paper, the authors prove to obtain the neighborhood point operator of the quadratic hinge loss function

$$\mathrm{prox}_{\ell_{\mathrm{sh}}}(s) = \begin{cases} \dfrac{s}{1+2\alpha}, & s \geqslant 0; \\ s, & s < 0. \end{cases}$$

The image of its proximity point operator is shown in Fig. 11(a). Unlike the proximity operator of the hinge loss function, when $s \geqslant 0$, its proximity operator is $\frac{s}{1+2\alpha}$.

(6) (Huber hinge loss function) In this paper, the authors prove to obtain the neighborhood point operator of the Huber hinge loss function

$$\mathrm{prox}_{\ell_{\mathrm{hh}}}(s) = \begin{cases} s-\alpha, & s \geqslant \alpha+\delta; \\ \dfrac{\delta s}{\alpha+\delta}, & s \in [0, \alpha+\delta); \\ s, & s < 0, \end{cases}$$

where $\delta > 0$. The image of its neighborhood point operator is shown in Fig. 11(a). Unlike the neighborhood point operator of the hinge loss function, when $s \in [0, \alpha+\delta)$, its neighborhood point operator is $\frac{\delta s}{\alpha+\delta}$.

(7) (Least squares loss function) In 1993, Frank and Friedman [23] gave the neighborhood point operator for the least squares loss function

$$\mathrm{prox}_{\ell_{\mathrm{ls}}}(s) = \frac{s}{1+2\alpha}, \quad s \in \mathbb{R}.$$

The image of its proximity point operator is shown in Fig. 11(b). Unlike the neighborhood point operator of the quadratic hinge loss function, when $s \leqslant 0$, its neighborhood point operator is $\frac{s}{1+2\alpha}$.

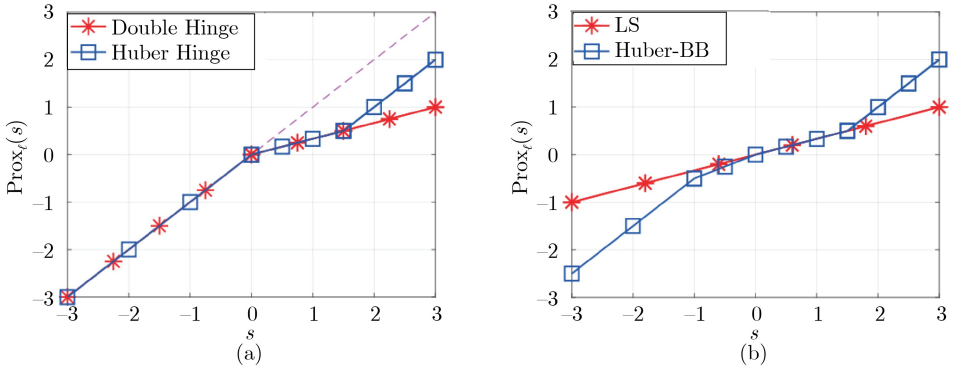

Fig. 11 Schematic diagram of the neighborhood point operator of the convex smooth loss function

(8) (Huber bouncing ball loss function) In this paper, the authors prove that the neighborhood point operator of the Huber bouncing ball loss function is obtained

$$\operatorname{prox}_{\ell_{\mathrm{hp}}}(s)=\begin{cases}s-\alpha, & s\geqslant\alpha+\delta;\\ \dfrac{\delta s}{\delta+\alpha}, & s\in(0,\alpha+\delta);\\ \dfrac{\delta s}{\delta+\tau\alpha}, & s\in(-(\tau\alpha+\delta),0];\\ s+\tau\alpha, & s\leqslant-(\tau\alpha+\delta),\end{cases}$$

where $\tau\in[0,1]$, $\delta>0$. The image of its neighborhood point operator is shown in Fig. 11(b). Unlike the neighborhood operator of the bouncing ball loss function, when $\in(0,\alpha+\delta)$, its neighborhood operator is $\frac{\delta s}{\delta+\alpha}$; when $\in(-(\tau\alpha+\delta),0]$, its neighborhood operator is $\frac{\delta s}{\delta+\tau\alpha}$.

3.2.3 Neighborhood point operator for nonconvex nonsmooth loss function

(9) (Slipway loss function) In 2020, Wang et al. [61] gave two different proximity point operators for $\alpha\in(0,2)$ and $\alpha\geqslant 2$ for the chute loss function. When $\alpha\in(0,2)$, the proximity operator of the slideway loss function is

$$\operatorname{prox}_{\ell_{\mathrm{rl}}}(s)=\begin{cases}s, & s>1+\dfrac{\alpha}{2};\\ s \text{ or } s-\alpha, & s=1+\dfrac{\alpha}{2};\\ s-\alpha, & s\in\left[\alpha,1+\dfrac{\alpha}{2}\right);\\ 0, & s\in(0,\alpha);\\ s, & s\leqslant 0.\end{cases}$$

The image of its proximity point operator is shown in Fig. 12(a). Unlike the proximity operator of the hinge loss function, when $s=1+\frac{\alpha}{2}$, its proximity operator is s or $s-\alpha$; when $s>1+\frac{\alpha}{2}$, its proximity operator is s.

When $\alpha\geqslant 2$, the neighborhood operator of the sliding loss function is

$$\operatorname{prox}_{\ell_{\mathrm{rl}}}(s)=\begin{cases}s, & s>\sqrt{2\alpha};\\ s \text{ or } 0, & s=\sqrt{2\alpha};\\ 0, & s\in[0,\sqrt{2\alpha});\\ s, & s<0.\end{cases}$$

The image of its proximity point operator is shown in Fig. 12(a). Unlike the neighborhood point operator of the hinge loss function, when $s\in[0,\sqrt{2\alpha})$, its neighborhood point operator is 0; when $s=\sqrt{2\alpha}$, its neighborhood point operator is s or 0; when $s>\sqrt{2\alpha}$, its neighborhood point operator is s.

(10) (Truncated least squares loss function) In this paper, the authors prove that the neighborhood operator of truncated least squares loss function

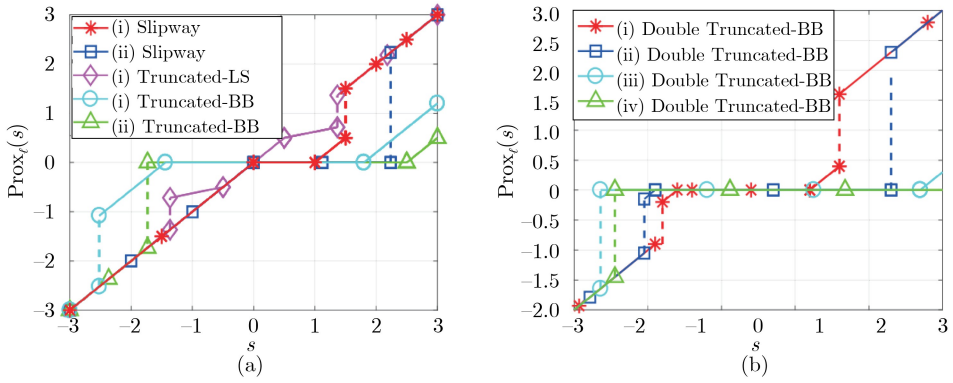

Fig. 12 Non-convex non-smooth loss function of the neighborhood point operator, where the multi-valued points are indicated by dashed lines

$$\mathrm{prox}_{\ell_{\mathrm{tll}}}(s)=\begin{cases} s, & |s|>\sqrt{2\alpha+1}(\mu-\varepsilon)+\varepsilon;\\ s \text{ or } \dfrac{\mathrm{sgn}(s)(|s|+2\alpha\varepsilon)}{2\alpha+1}, & |s|=\sqrt{2\alpha+1}(\mu-\varepsilon)+\varepsilon;\\ \dfrac{\mathrm{sgn}(s)(|s|+2\alpha\varepsilon)}{2\alpha+1}, & |s|\in(\varepsilon,\sqrt{2\alpha+1}(\mu-\varepsilon)+\varepsilon);\\ s, & |s|\in[0,\varepsilon], \end{cases}$$

where $0<\varepsilon<\mu$. Its neighborhood point operator is shown in Fig. 12(a). Unlike the neighborhood operator of the least-squares loss function, when $|s|\in[0,\varepsilon]$, its neighborhood operator is s; when $|s|\in(\varepsilon,\sqrt{2\alpha+1}(\mu-\varepsilon)+\varepsilon)$, its neighborhood operator is $\frac{\mathrm{sgn}(s)(|s|+2\alpha\varepsilon)}{2\alpha+1}$; when $|s|=\sqrt{2\alpha+1}(\mu-\ \varepsilon)+\varepsilon$, its neighborhood operator is s or $\frac{\mathrm{sgn}(s)(|s|+2\alpha\varepsilon)}{2\alpha+1}$. when $|s|>\sqrt{2\alpha+1}(\mu-\ \varepsilon)+\varepsilon$, its neighborhood operator is s.

(11) (Truncated bouncing ball loss function) In this paper, the authors prove two different proximity operators for $\alpha\in(0,\frac{2\kappa}{\tau})$ and $\alpha\geqslant\frac{2\kappa}{\tau}$ to truncate the bouncing ball loss function.

When $\alpha\in(0,\frac{2\kappa}{\tau})$, the proximity operator of the truncated bouncing ball loss function is

$$\mathrm{prox}_{\ell_{\mathrm{tp}}}(s)=\begin{cases} s-\alpha, & s\geqslant\alpha;\\ 0, & s\in[-\tau\alpha,\alpha);\\ s+\tau\alpha, & s\in\left(-\dfrac{\tau\alpha}{2}-\kappa,-\tau\alpha\right);\\ s+\tau\alpha \text{ or } s, & s=-\dfrac{\tau\alpha}{2}-\kappa;\\ s, & s<-\dfrac{\tau\alpha}{2}-\kappa, \end{cases}$$

where $\tau\in[0,1]$, $\kappa>0$. Its proximity point operator is shown in Fig. 12(a). Unlike the neighborhood operator of the bouncing ball loss function, when $s=-\frac{\tau\alpha}{2}-\kappa$, its neighborhood operator is $s+\tau\alpha$ or s; when $s<-\frac{\tau\alpha}{2}-\kappa$, its neighborhood operator is s. When $\alpha\geqslant\frac{2\kappa}{\tau}$, the neighborhood operator of the truncated bouncing ball loss function is

$$\operatorname{prox}_{\ell_{\mathrm{tp}}}(s)=\begin{cases}s-\alpha, & s\geqslant\alpha;\\ 0, & s\in(-\sqrt{2\alpha\tau\kappa},\alpha);\\ 0 \text{ or } s, & s=-\sqrt{2\alpha\tau\kappa};\\ s, & s<-\sqrt{2\alpha\tau\kappa},\end{cases}$$

where $\tau\in[0,1]$, $\kappa>0$. Its proximity point operator is shown in Fig. 12(a). Unlike the neighborhood operator of the bouncing ball loss function, when $s\in(-\sqrt{2\alpha\tau\kappa},\alpha)$, its neighborhood operator is 0; when $s=-\sqrt{2\alpha\tau\kappa}$, its neighborhood operator is 0 or s; when $s<-\sqrt{2\alpha\tau\kappa}$, its neighborhood operator is s.

(12) (Double truncated bouncing ball loss function) In this paper, the authors prove four different proximity operators for the double truncated bouncing ball loss function for different values of parameters τ,κ,μ.

(i) When $\alpha\in(0,\frac{2\kappa}{\tau})$ and $\alpha\in(0,2\mu)$, the proximity operator of the double-truncated bouncing ball loss function is

$$\operatorname{prox}_{\ell_{\mathrm{btp}}}(s)=\begin{cases}s, & s>\mu+\dfrac{\alpha}{2};\\ s \text{ or } s-\alpha, & s=\mu+\dfrac{\alpha}{2};\\ s-\alpha, & s\in\left[\alpha,\mu+\dfrac{\alpha}{2}\right);\\ 0, & s\in(-\tau\alpha,\alpha);\\ s+\tau\alpha, & s\in\left(-\dfrac{\tau\alpha}{2}-\kappa,-\tau\alpha\right];\\ s+\tau\alpha \text{ or } s, & s=-\dfrac{\tau\alpha}{2}-\kappa;\\ s, & s<-\dfrac{\tau\alpha}{2}-\kappa,\end{cases}$$

where $\tau\in[0,1]$, $\kappa,\mu>0$. The image of its neighborhood point operator is shown in Fig. 12(b). Unlike the truncated bouncing ball loss function neighborhood operator of $\alpha\in(0,\frac{2\kappa}{\tau})$, when $s=\mu+\frac{\alpha}{2}$, the neighborhood operator is s or $s-\alpha$; when $s>\mu+\frac{\alpha}{2}$. when $s>\mu+\frac{\alpha}{2}$, its neighborhood operator is s.

(ii) When $\alpha\in(0,\frac{2\kappa}{\tau})$ and $\alpha\geqslant 2\mu$, the neighborhood point operator of the double truncated bouncing ball loss function is

$$\operatorname{prox}_{\ell_{\mathrm{btp}}}(s)=\begin{cases}s, & s>\sqrt{2\alpha\mu};\\ s \text{ or } 0, & s=\sqrt{2\alpha\mu};\\ 0, & s\in[-\tau\alpha,\sqrt{2\alpha\mu});\\ s+\tau\alpha, & s\in\left(-\dfrac{\tau\alpha}{2}-\kappa,-\tau\alpha\right);\\ s+\tau\alpha \text{ or } s, & s=-\dfrac{\tau\alpha}{2}-\kappa;\\ s, & s<-\dfrac{\tau\alpha}{2}-\kappa,\end{cases}$$

where $\tau \in [0,1]$, $\kappa, \mu > 0$. The image of its neighborhood point operator is shown in Fig. 12(b). Unlike the truncated bouncing ball loss function neighborhood operator of $\alpha \in (0, \frac{2\kappa}{\tau})$, when $s \in [-\tau\alpha, \sqrt{2\alpha\mu})$, the neighborhood operator is 0; when $s = \sqrt{2\alpha\mu}$, the neighborhood operator is $s \in [-\tau\alpha, \sqrt{2\alpha\mu}]$. When $s = \sqrt{2\alpha\mu}$, its neighborhood operator is s or 0; when $s > \sqrt{2\alpha\mu}$, its neighborhood operator is s.

(iii) When $\alpha \geqslant \frac{2\kappa}{\tau}$ and $\alpha \in (0, 2\mu)$, the neighborhood operator of the double-truncated bouncing ball loss function is

$$\mathrm{prox}_{\ell_{\mathrm{btp}}}(s) = \begin{cases} s, & s > \mu + \dfrac{\alpha}{2}; \\ s \text{ or } s-\alpha, & s = \mu + \dfrac{\alpha}{2}; \\ s-\alpha, & s \in \left[\alpha, \mu + \dfrac{\alpha}{2}\right); \\ 0, & s \in (-\sqrt{2\alpha\tau\kappa}, \alpha); \\ 0 \text{ or } s, & s = -\sqrt{2\alpha\tau\kappa}; \\ s, & s < -\sqrt{2\alpha\tau\kappa}, \end{cases}$$

where $\tau \in [0,1]$, $\kappa, \mu > 0$. The image of its neighborhood point operator is shown in Fig. 12(b). Unlike the truncated bouncing ball loss function neighborhood operator of $\alpha \geqslant \frac{2\kappa}{\tau}$, when $s = \mu + \frac{\alpha}{2}$, its neighborhood operator is s or $s-\alpha$; when $s > \mu + \frac{\alpha}{2}$, its neighborhood operator is s. When $s > \mu + \frac{\alpha}{2}$, its neighborhood operator is s.

(iv) When $\alpha \geqslant \frac{2\kappa}{\tau}$ and $\alpha \geqslant 2\mu$, the neighborhood operator of the double-truncated bouncing ball loss function is

$$\mathrm{prox}_{\ell_{\mathrm{btp}}}(s) = \begin{cases} s, & s > \sqrt{2\alpha\mu}; \\ s \text{ or } 0, & s = \sqrt{2\alpha\mu}; \\ 0, & s \in (-\sqrt{2\alpha\tau\kappa}, \sqrt{2\alpha\mu}); \\ 0 \text{ or } s, & s = -\sqrt{2\alpha\tau\kappa}; \\ s, & s < -\sqrt{2\alpha\tau\kappa}, \end{cases}$$

where $\tau \in [0,1]$, $\kappa, \mu > 0$. The image of its neighborhood point operator is shown in Fig. 12(b). Unlike the truncated bouncing ball loss function neighborhood operator of $\alpha \geqslant \frac{2\kappa}{\tau}$, when $s \in [-\sqrt{2\alpha\tau\kappa}, \sqrt{2\alpha\mu})$, its neighborhood operator is 0; when $s = \sqrt{2\alpha\mu}$, its neighborhood operator is s or 0; when $s > \sqrt{2\alpha\mu}$, its neighborhood operator is s.

3.3 Fenchel covariance

In this subsection, we introduce and give the Fenchel covariance of 18 SVM agent loss functions. The definition of Fenchel covariance is given first.

Definition 3.3 [3] Let function $f : \mathbb{R} \to \overline{\mathbb{R}}$, its Fenchel conjugate $f^* : \mathbb{R} \to [-\infty, +\infty]$ is defined as

$$f^*(t^*) := \sup_{t\in\mathbb{R}}\{t^*t - f(t)\}.$$

It is worth noting that whether the function $f(t)$ is convex or not, its Fenchel conjugate function $f^*(t^*)$ must be a closed convex function.

3.3.1 Fenchel conjugate of convex nonsmooth loss function

(1) (Hinge loss function) In 1995, Cortes and Vapnik [16] gave the Fenchel conjugate of the hinge loss function

$$\ell^*_{\mathrm{hl}}(t^*) = \begin{cases} 0, & t^* \in [0,1]; \\ +\infty, & \text{other}. \end{cases}$$

When $t^* \in [0,1]$, its Fenchel conjugate is 0; when t^* takes other cases, its Fenchel conjugate is $+\infty$.

(2) (Generalized hinge loss function) In this paper, the authors prove that the Fenchel conjugate of the generalized hinge loss function

$$\ell^*_{\mathrm{gh}}(t^*) = \begin{cases} t^* - 1, & t^* \in (1,\eta]; \\ 0, & t^* \in [0,1]; \\ +\infty, & \text{other}, \end{cases}$$

where $\eta > 1$. Unlike the Fenchel conjugate of the hinge loss function, when $t^* \in (1,\eta]$, its Fenchel conjugate is $t^* - 1$.

(3) (Bouncing ball loss function) In 2013, Jumutc et al. [30] gave the Fenchel conjugate of the bouncing ball loss function

$$\ell^*_{\mathrm{pl}}(t^*) = \begin{cases} 0, & t^* \in [-\tau,1]; \\ +\infty, & \text{other}, \end{cases}$$

where $\tau \in [0,1]$. Unlike the Fenchel conjugate of the hinge loss function, when $t^* \in [-\tau,0)$, its Fenchel conjugate is 0.

(4) (ε-insensitive bouncing ball loss function) In 2014, Huang et al. [26] gave the Fenchel conjugate of ε-insensitive bouncing ball loss function

$$\ell^*_{\mathrm{ip}}(t^*) = \begin{cases} t^*\varepsilon, & t^* \in [0,1]; \\ -\dfrac{t^*\varepsilon}{\tau}, & t^* \in [-\tau,0); \\ +\infty, & \text{other}, \end{cases}$$

where $\tau \in [0,1]$, $\varepsilon > 0$. Unlike the Fenchel conjugate of the bouncing ball loss function, when $t^* \in [0,1]$, its Fenchel conjugate is $t^*\varepsilon$; when $t^* \in [-\tau,0)$, its Fenchel conjugate is $-\frac{t^*\varepsilon}{\tau}$.

3.3.2 Fenchel conjugate of convex smooth loss function

(5) (Double hinge loss function) In this paper, the authors prove that the Fenchel conjugate of the quadratic hinge loss function is obtained

$$\ell^*_{\mathrm{sh}}(t^*) = \begin{cases} \dfrac{(t^*)^2}{4}, & t^* \geqslant 0; \\ +\infty, & \text{other}. \end{cases}$$

Unlike the Fenchel conjugate of the hinge loss function, when $t^* \geqslant 0$, its Fenchel conjugate is $\frac{(t^*)^2}{4}$.

(6) (Huber hinge loss function) In this paper, the authors prove that the Fenchel conjugate of the Huber hinge loss function is obtained

$$\ell^*_{\rm hh}(t^*) = \begin{cases} \dfrac{\delta(t^*)^2}{2}, & t^* \in [0,1]; \\ +\infty, & \text{other}, \end{cases}$$

where $\delta > 0$. Unlike the Fenchel conjugate of the hinge loss function, when $t^* \in [0,1]$, its Fenchel conjugate is $\frac{\delta(t^*)^2}{2}$.

(7) (Logarithmic loss function) In this paper, the authors prove that the Fenchel conjugate of the logarithmic loss function is obtained

$$\ell^*_{\rm ll}(t^*) = \begin{cases} t^*\log(t^*) + (1-t^*)\log(1-t^*) + t^*, & t^* \in (0,1); \\ 0, & t^* = 0; \\ 1, & t^* = 1; \\ +\infty, & \text{other}. \end{cases}$$

Unlike the Fenchel conjugate of the hinge loss function, when $t^* \in (0,1)$, its Fenchel conjugate is $t^*\log(t^*) + (1-t^*)\log(1-t^*) + t^*$; when $t^* = 0$, its Fenchel conjugate is 0; when $t^* = 1$. the Fenchel conjugate is 1.

(8) (Least squares loss function) In 2000, Suykens and Vandewalle [57] gave the Fenchel conjugate of the least squares loss function

$$\ell^*_{\rm ls}(t^*) = \frac{(t^*)^2}{4}, \quad t^* \in \mathbb{R}.$$

Unlike the Fenchel conjugate of the quadratic hinge loss function, when $t^* < 0$, its Fenchel conjugate is $\frac{(t^*)^2}{4}$.

(9) (Huber bouncing ball loss function) In this paper, the authors prove that the Fenchel conjugate of the Huber bouncing ball loss function is obtained

$$\ell^*_{\rm hp}(t^*) = \begin{cases} \dfrac{\delta(t^*)^2}{2}, & t^* \in [0,1]; \\ \dfrac{\delta(t^*)^2}{2\tau}, & t^* \in [-\tau, 0); \\ +\infty, & \text{other}, \end{cases}$$

where $\tau \in [0,1]$, $\delta > 0$. Unlike the Fenchel conjugate of the bouncing ball loss function, when $t^* \in [0,1]$, its Fenchel conjugate is $\frac{\delta(t^*)^2}{2}$; when $t^* \in [-\tau, 0)$, its Fenchel conjugate is $\frac{\delta(t^*)^2}{2\tau}$.

3.3.3 Fenchel conjugate of nonconvex nonsmooth loss function

(10) (Slideway loss function) In this paper, the authors prove to obtain the Fenchel conjugate of the slideway loss function

$$\ell^*_{\rm rl}(t^*) = \begin{cases} 0, & t^* = 0; \\ +\infty, & \text{other}. \end{cases}$$

(11) (Truncated logarithmic loss function) In this paper, the authors prove

that the Fenchel conjugate of the truncated logarithmic loss function is obtained

$$\ell^*_{\mathrm{tll}}(t^*) = \begin{cases} 0, & t^* = 0; \\ +\infty, & \text{other.} \end{cases}$$

(12) (Truncated least squares loss function) In this paper, the authors prove that the Fenchel conjugate of the truncated least squares loss function is obtained

$$\ell^*_{\mathrm{tll}}(t^*) = \begin{cases} 0, & t^* = 0; \\ +\infty, & \text{other.} \end{cases}$$

(13) (Truncated bouncing ball loss function) In this paper, the authors prove that the Fenchel conjugate of the truncated bouncing ball loss function is obtained

$$\ell^*_{\mathrm{tp}}(t^*) = \begin{cases} 0, & t^* \in [0,1]; \\ +\infty, & \text{other.} \end{cases}$$

(14) (Double truncated bouncing ball loss function) In this paper, the authors prove that the Fenchel conjugate of the double truncated bouncing ball loss function is obtained

$$\ell^*_{\mathrm{btp}}(t^*) = \begin{cases} 0, & t^* = 0; \\ +\infty, & \text{other.} \end{cases}$$

Among the above five non-convex and non-smooth loss functions, the truncated bouncing ball loss function is bounded in the negative half-axis and unbounded in the positive half-axis. When $t^* \in [0,1]$, the Fenchel conjugate of the truncated spherical loss function is 0; when t^* takes other cases, the Fenchel conjugate is $+\infty$. The other four loss functions are bounded in the positive and negative semi-axes, and they have the same Fenchel conjugate. When $t^* = 0$, their Fenchel conjugate is 0; when t^* is taken as other cases, their Fenchel conjugate is $+\infty$.

3.3.4 Fenchel conjugate of non-convex smooth loss function

(15) (Generalized exponential loss function) In this paper, the authors prove that the Fenchel conjugate of the generalized exponential loss function is obtained

$$\ell^*_{\mathrm{gel}}(t^*) = \begin{cases} 0, & t^* = 0; \\ +\infty, & \text{other.} \end{cases}$$

(16) (Generalized logarithmic loss function) In this paper, the authors prove that the Fenchel conjugate of the generalized logarithmic loss function is obtained

$$\ell^*_{\mathrm{gll}}(t^*) = \begin{cases} 0, & t^* = 0; \\ +\infty, & \text{other.} \end{cases}$$

(17) (Sigmoid loss function) In this paper, the authors prove that the Fenchel

conjugate of the Sigmoid loss function is obtained

$$\ell^*_{\mathrm{sl}}(t^*) = \begin{cases} 0, & t^* = 0; \\ +\infty, & \text{other}. \end{cases}$$

(18) (Cumulative distribution loss function) In this paper, the authors prove that the Fenchel conjugate of the cumulative distribution loss function is obtained

$$\ell^*_{\mathrm{cd}}(t^*) = \begin{cases} 0, & t^* = 0; \\ +\infty, & \text{other}. \end{cases}$$

The above four non-convex smooth loss functions are bounded loss functions in both positive and negative half-axes, and they have the same Fenchel conjugate. When $t^* = 0$, their Fenchel conjugate is 0; when t^* is taken as other cases, their Fenchel conjugate is $+\infty$.

4 0-1 loss functions and their variational properties

In the previous section we give the subdifferentiation of 0-1 loss function for 18 commonly used agent loss functions, the neighborhood point operator and Fenchel conjugate. In this section we introduce and give the 0-1 loss function and its three variational properties.

In 1995, Cortes and Vapnik [16] pointed out that the 0-1 loss function is the most desirable loss function for soft interval SVMs, and its mathematical expression is

$$\ell_{0/1}(t) = \begin{cases} 1, & t > 0; \\ 0, & t \leqslant 0. \end{cases}$$

The function image is shown in Fig. 13(a). The 0-1 loss function portrays the discrete nature [25, 38] of the binary classification problem of judging only yes or no. For $t \leqslant 0$ samples, the loss value is 0; for $t > 0$ samples, the loss value is 1, and these samples are non-support vectors [62]. Therefore, it is sparse and robust to outliers.

In 2019, Zhang [72] gave the Clarke subdifferential of the 0-1 loss function

$$\partial^C \ell_{0/1}(t) \begin{cases} \geqslant 0, & t = 0; \\ = 0, & t \neq 0. \end{cases}$$

Its Clarke subdifferential image is shown in Fig. 13(b). When $t = 0$, its Clarke subdifferential is greater than or equal to 0; when $t \neq 0$, its gradient is 0.

In 2019, Wang et al. [62] gave the neighborhood point operator for the 0-1 loss function

$$\mathrm{prox}_{\ell_{0/1}}(s) = \begin{cases} s, & s > \sqrt{2\alpha}; \\ s \text{ or } 0, & s = \sqrt{2\alpha}; \\ 0, & s \in (0, \sqrt{2\alpha}); \\ s, & s \leqslant 0, \end{cases}$$

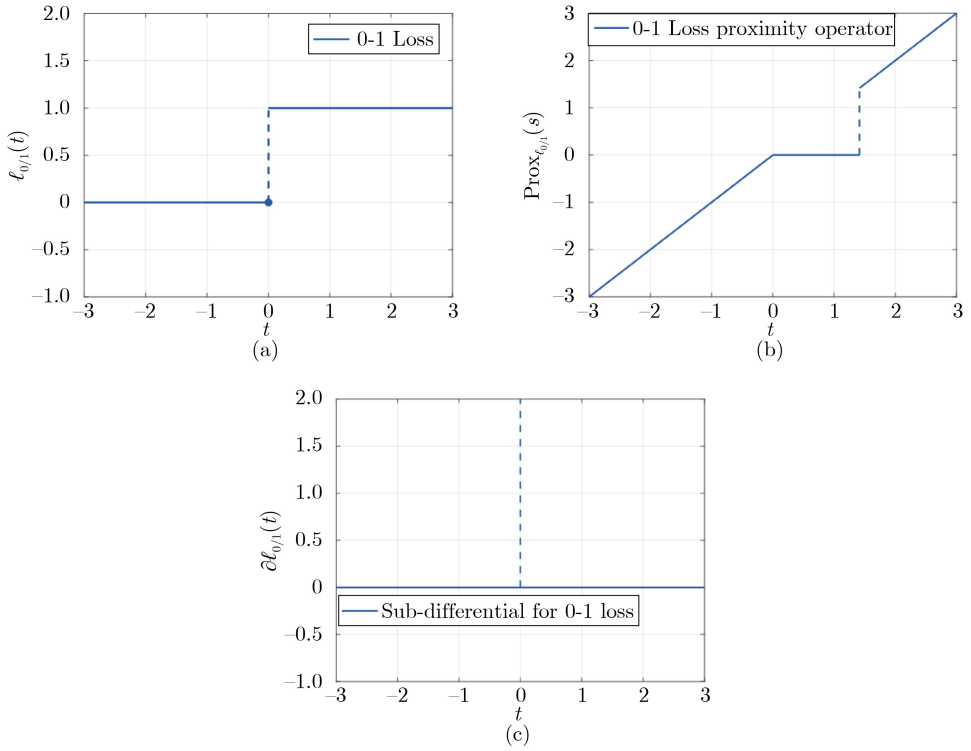

Fig. 13 (a) Schematic diagram of the 0-1 loss function, where the discontinuities are indicated by dashed lines; (b) Schematic diagram of the Clarke subdifferential of the 0-1 loss function, where the set of Clarke subdifferentials at the non-differentiable points is shown by the dashed line; (c) Schematic diagram of the 0-1 loss function for the neighborhood operator, where the multi-valued points are shown by dashed lines

whose neighborhood point operator image is shown in Fig. 13(c). When $s > \sqrt{2\alpha}$ or $s \leqslant 0$, its neighborhood operator is s; when $s = \sqrt{2\alpha}$, its neighborhood operator is s or 0; when $s \in (0, \sqrt{2\alpha})$, its neighborhood operator is 0.

In this paper, the authors prove that the Fenchel conjugate of the 0-1 loss function

$$\ell_{0/1}^*(t^*) = \begin{cases} 0, & t^* = 0; \\ +\infty, & \text{other}. \end{cases}$$

When $t^* = 0$, its Fenchel conjugate is 0; when t^* takes other cases, its Fenchel conjugate is $+\infty$.

5 Conclusion

In this paper, we summarize the 0-1 loss function and its 18 commonly used SVM proxy loss functions, point out the reasons and advantages and disadvantages of each loss function, and give three important variational properties: subdifferential, neighborhood operator and Fenchel conjugate. In order to facilitate a quick reference and comparison, the 19 loss functions and their properties are summarized in Table 1. We hope that this paper can inspire readers to study

Table 1 Properties of the 19 loss functions in the quad SVM

("Y" means the loss function has the corresponding property, "N" means the loss function does not have the corresponding property, "[*]" means that the corresponding property of the loss function is given by this paper)

Loss function	Convexity	Boundedness	Sparsity	Robustness	Subdifferential	Neighborhood point operator	Fenchel conjugate
0-1 loss [62]	N	Y	Y	Y	Y	Y [72]	Y [62]
	Y[*]						
Hinge loss [16]	Y	N	Y	N	Y [16]	Y [67]	Y [16]
Generalized hinge loss [2]	Y	N	Y	N	Y [2]	Y[*]	Y[*]
Pinball loss [30]	Y	N	N	N	Y [30]	Y[*]	Y [30]
ε-insensitive bouncing ball loss [26]	Y	N	Y	N	Y [26]	Y[*]	Y [26]
Secondary hinge loss [16]	Y	N	Y	N	Y [16]	Y[*]	Y[*]
Huber hinge loss [10]	Y	N	Y	N	Y [10]	Y[*]	Y[*]
Logarithmic loss [60]	Y	N	N	N	Y [60]	N	Y[*]
Least squares loss [57]	Y	N	N	N	Y [57]	Y [23]	Y [57]
Huber bouncing ball loss [78]	Y	N	N	N	Y [78]	Y[*]	Y[*]
Chute loss [54]	N	Y	Y	Y	Y [54]	Y [61]	Y[*]
Truncated logarithmic loss [46]	N	Y	Y	Y	Y [46]	N	Y[*]
Truncated least squares loss [42]	N	Y	Y	Y	Y	Y [42]	Y[*]
	Y[*]						
Truncated bouncing ball loss [55]	N	N	Y	N	Y [55]	Y[*]	Y[*]
Double truncated bouncing ball loss [68]	N	Y	Y	Y	Y [68]	Y[*]	Y[*]
Generalized exponential loss [22]	N	Y	Y	Y	Y [22]	N	Y[*]
Generalized logarithmic loss [22]	N	Y	Y	Y	Y	Y [22]	N
	Y[*]						
Sigmoid loss [47]	N	Y	N	Y	Y [47]	N	Y[*]
Cumulative distribution loss [24]	N	Y	N	Y	Y [24]	N	Y[*]

SVM models and propose new solution algorithms, and promote the development of SVM.

References

1. Akay M F. Support vector machines combined with feature selection for breast cancer diagnosis. Expert Syst Appl, 2009, 36(2): 3240–3247
2. Bartlett P L, Wegkamp M H. Classification with a reject option using a hinge loss. J Mach Learn Res, 2008, 9: 1823–1840

3. Beck A. First-Order Methods in Optimization. MOS-SIAM Series on Optimization, Vol 25. Philadelphia, PA: SIAM, 2017
4. Brooks J P. Support vector machines with the ramp loss and the hard margin loss. Oper Res, 2011, 59(2): 467–479
5. Cao L J, Keerthi S S, Ong C J, Zhang J Q, Periyathamby U, Fu X J, Lee H P. Parallel sequential minimal optimization for the training of support vector machines. IEEE Trans Neural Netw, 2006, 17(4): 1039–1049
6. Carrizosa E, Nogales-Gómez A, Romero Morales D. Heuristic approaches for support vector machines with the ramp loss. Optim Lett, 2014, 8(3): 1125–1135
7. Chang C-C, Hsu C-W, Lin C-J. The analysis of decomposition methods for support vector machines. IEEE Trans Neural Netw, 2000, 11(4): 1003–1008
8. Chang C-C, Lin C-J. LIBSVM: A library for support vector machines. ACM Trans Intell Syst Technol, 2011, 2(3): 27
9. Chang K-W, Hsieh C-J, Lin C-J. Coordinate descent method for large-scale L2-loss linear support vector machines. J Mach Learn Res, 2008, 9: 1369–1398
10. Chapelle O. Training a support vector machine in the primal. Neural Comput, 2007, 19(5): 1155–1178
11. Chapelle O, Haffner P, Vapnik V N. Support vector machines for histogram-based image classification. IEEE Trans Neural Netw, 1999, 10(5): 1055–1064
12. Chen H L, Yang B, Liu J, Liu D Y. A support vector machine classifier with rough set-based feature selection for breast cancer diagnosis. Expert Syst Appl, 2011, 38(7): 9014–9022
13. Clarke F H. Optimization and Nonsmooth Analysis. New York: John Wiley & Sons, 1983
14. Collobert R, Sinz F, Weston J, Bottou L. Trading convexity for scalability. In: ICML 2006, Proceedings of the 23rd International Conference on Machine Learning (Cohen W W, Moore A, eds). New York: Association for Computing Machinery, 2006, 201–208
15. Collobert R, Sinz F, Weston J, Bottou L. Large scale transductive SVMs. J Mach Learn Res, 2006, 7: 1687–1712
16. Cortes C, Vapnik V. Support vector networks. Mach Learn, 1995, 20(3): 273–297
17. De Kruif B J, De Vries T J A. Pruning error minimization in least squares support vector machines. IEEE Trans Neural Netw, 2003, 14(3): 696–702
18. Deng N Y, Tian Y J, Zhang C H. Support Vector Machines: Optimization Based Theory, Algorithms, and Extensions. Boca Raton, FL: CRC Press, 2012
19. Ertekin S, Bottou L, Giles C L. Nonconvex online support vector machines. IEEE Trans Pattern Anal Mach Intell, 2011, 33(2): 368–381
20. Fan R-E, Chang K-W, Hsieh C-J, Wang X-R, Lin C-J. LIBLINEAR: A library for large linear classification. J Mach Learn Res, 2008, 9: 1871–1874
21. Fan R-E, Chen P-H, Lin C-J. Working set selection using second order information for training support vector machines. J Mach Learn Res, 2005, 6: 1889–1918
22. Feng Y L, Yang Y N, Huang X L, Mehrkanoon S, Suykens J A K. Robust support vector machines for classification with nonconvex and smooth losses. Neural Comput, 2016, 28(6): 1217–1247
23. Frank I E, Friedman J H. A statistical view of some chemometrics regression tools. Technometrics, 1993, 35(2): 109–135
24. Ghanbari H, Li M H, Scheinberg K. Novel and efficient approximations for zero-one loss of linear classifiers, 2019, arXiv: 1903.00359
25. Huang L W, Shao Y H, Zhang J, Zhao Y T, Teng J Y. Robust rescaled hinge loss twin support vector machine for imbalanced noisy classification. IEEE Access, 2019, 7: 65390–65404
26. Huang X L, Shi L, Suykens J A K. Support vector machine classifier with pinball loss. IEEE Trans Pattern Anal Mach Intell, 2014, 36(5): 984–997

27. Huang X L, Shi L, Suykens J A K. Ramp loss linear programming support vector machine. J Mach Learn Res, 2014, 15: 2185−2211
28. Huang X L, Shi L, Suykens J A K. Sequential minimal optimization for SVM with pinball loss. Neurocomputing, 2015, 149(C): 1596−1603
29. Huang X L, Shi L, Suykens J A K. Solution path for pin-SVM classifiers with positive and negative τ values. IEEE Trans Neural Netw Learn Syst, 2017, 28(7): 1584−1593
30. Jumutc V, Huang X L, Suykens J A K. Fixed-size Pegasos for hinge and pinball loss SVM. In: The 2013 International Joint Conference on Neural Networks (IJCNN). Piscataway, NJ: IEEE, 2013
31. Keerthi S S, DeCoste D. A modified finite Newton method for fast solution of large scale linear SVMs. J Mach Learn Res, 2005, 6: 341−361
32. Keerthi S S, Gilbert E G. Convergence of a generalized SMO algorithm for SVM classifier design. Machine Learning, 2002, 46: 351−360
33. Keerthi S S, Shevade S K. SMO algorithm for least-squares SVM formulations. Neural Comput, 2003, 15(2): 487−507
34. Keerthi S S, Shevade S K, Bhattacharyya C, Murthy K R K. Improvements to Platt's SMO algorithm for SVM classifier design. Neural Comput, 2014, 13(3): 637−649
35. Khan N M, Ksantini R, Ahmad I S, Boufama B. A novel SVM+NDA model for classification with an application to face recognition. Pattern Recognition, 2012, 45(1): 66−79
36. Lee C-P, Lin C-J. A Study on L2-loss (squared hinge-loss) multiclass SVM. Neural Comput, 2013, 25(5): 1302−1323
37. Lee Y-J, Mangasarian O L. SSVM: a smooth support vector machine for classification. Comput Optim Appl, 2001, 20(1): 5−22
38. Li H. Statistical Learning Methods, 2nd ed. Beijing: Tsinghua Univ Press, 2019 (in Chinese)
39. Li J T, Jia Y M, Li W L. Adaptive huberized support vector machine and its application to microarray classification. Neural Computing and Applications, 2011, 20(1): 123−132
40. Lin C-J. On the convergence of the decomposition method for support vector machines. IEEE Trans Neural Netw, 2001, 12(6): 1288−1298
41. Lin C-J. Asymptotic convergence of an SMO algorithm without any assumptions. IEEE Trans Neural Netw, 2002, 13(1): 248−250
42. Liu D L, Shi Y, Tian Y J, Huang X K. Ramp loss least squares support vector machine. J Comput Sci, 2016, 14: 61−68
43. López J, Suykens J A K. First and second order SMO algorithms for LS-SVM classifiers. Neural Process Lett, 2011, 33(1): 31−44
44. Mančev D. A sequential dual method for the structured ramp loss minimization. Facta Univ Ser Math Inform, 2005, 30(1): 13−27
45. Mason L, Bartlett P L, Baxter J. Improved generalization through explicit optimization of margins. Mach Learn, 2000, 38(3): 243−255
46. Park S Y, Liu Y F. Robust penalized logistic regression with truncated loss functions. Canad J Statist, 2011, 39(2): 300−323
47. Pérez-Cruz F, Navia-Vázquez A, Figueiras-Vidal A R, Artés- Rodríguez A. Empirical risk minimization for support vector classifiers. IEEE Trans Neural Netw, 2003, 14(2): 296−303
48. Rockafellar R T, Wets R J-B. Variational Analysis, Corrected 3rd printing. Grundlehren der Mathematischen Wissenschaften, Vol 317. Berlin: Springer-Verlag, 2009
49. Sabbah T, Ayyash M, Ashraf M. Hybrid support vector machine based feature selection method for text classification. The International Arab Journal of Information

Technology, 2018, 15(3A): 599–609
50. Shalev-Shwartz S, Singer Y, Srebro N, Cotter A. Pegasos: primal estimated subgradient solver for SVM. Math Program, 2011, 127(1): Ser B, 3–30
51. Shao Y H, Liu L M, Huang L W, Deng N Y. Key issues of support vector machines and future prospects. Sci Sin Math, 2020, 50(9): 1233–1248 (in Chinese)
52. Shao Y H, Yang K L, Liu M Z, Wang Z, Li C N, Chen W J. From support vector machine to nonparallel support vector machine. Operations Research Transactions, 2018, 22(2): 55–65(inChinese) (in Chinese)
53. Sharif W, Yanto I T R, Samsudin N A, Deris M M, Khan A, Mushtaq M F, Ashraf M. An optimised support vector machine with Ringed Seal Search algorithm for efficient text classification. Journal of Engineering Science and Technology, 2019, 14(3): 1601–1613
54. Shen X T, Tseng G C, Zhang X G, Wong W H. On ψ-learning. J Amer Statist Assoc, 2003, 98(463): 724–734
55. Shen X, Niu L F, Qi Z Q, Tian Y J. Support vector machine classifier with truncated pinball loss. Pattern Recognition, 2017, 68: 199–210
56. Steinwart I, Christmann A. Support Vector Machines. New York: Springer, 2008
57. Suykens J A K, Vandewalle J. Least squares support vector machine classifiers. Neural Process Lett, 1999, 9(3): 293–300
58. Tanveer M, Sharma S, Rastogi R, Anand P. Sparse support vector machine with pinball loss. Trans Emerg Telecommun Technol, 2021, 32(2): e3820
59. VenkateswarLal P, Nitta G R, Prasad A. Ensemble of texture and shape descriptors using support vector machine classification for face recognition. J Ambient Intell Humaniz Comput, 2019, https://doi:10.1007/s12652-019-01192-7, in press
60. Wahba G. Support vector machines, reproducing kernel Hilbert spaces, and randomized GACV. In: Advances in Kernel Methods—Support Vector Learning (Schölkopf B, Burges C J C, Smola A J, eds). Cambridge, MA: MIT Press, 1998, 69–88
61. Wang H J, Shao Y H, Xiu N H. Proximal operator and optimality conditions for ramp loss SVM. Optim Lett, 2022, 16: 999–1014
62. Wang H J, Shao Y H, Zhou S L, Zhang C, Xiu N H. Support vector machine classifier via L0/1 soft-margin loss. IEEE Trans Pattern Anal Mach Intell, 2022, 44(10): 7253–7265
63. Wang Q, Ma Y, Zhao K, Tian Y J. A comprehensive survey of loss functions in machine learning. Ann Data Sci, 2020, 9: 187–212
64. Wu Y C, Liu Y F. Robust truncated hinge loss support vector machines. J Amer Statist Assoc, 2007, 102(479): 974–983
65. Xu J M, Li L. A face recognition algorithm based on sparse representation and support vector machine. Computer Technology and Development, 2018, 28(2): 59–63(inChinese) (in Chinese)
66. Xu Y Y, Akrotirianakis I, Chakraborty A. Proximal gradient method for huberized support vector machine. Pattern Anal Appl, 2016, 19(4): 989–1005
67. Yan Y Q, Li Q N. An efficient augmented Lagrangian method for support vector machine. Optim Methods Softw, 2020, 35(4): 855–883
68. Yang L M, Dong H W. Support vector machine with truncated pinball loss and its application in pattern recognition. Chemometrics Intell Lab Syst, 2018, 177: 89–99
69. Yang Y, Zou H. An efficient algorithm for computing the HHSVM and its generalizations. J Comput Graph Statist, 2013, 22(2): 396–415
70. Yang Z J, Xu Y T. A safe accelerative approach for pinball support vector machine classifier. Knowledge-Based Syst, 2018, 147: 12–24
71. Yin J, Li Q N. A semismooth Newton method for support vector classification and regression. Comput Optim Appl, 2019, 73(2): 477–508
72. Zhang C. Support Vector Machine Classifier via 0-1 Loss Function. MS Thesis.

Beijing: Beijing Jiaotong University, 2019 (in Chinese)
73. Zhang T, Oles F J. Text categorization based on regularized linear classification methods. Information Retrieval, 2001, 4(1): 5–31
74. Zhang W, Yoshida T, Tang X J. Text classification based on multi-word with support vector machine. Knowledge-Based Syst, 2008, 21(8): 879–886
75. Zhao L, Mammadov M, Yearwood J. From convex to nonconvex: a loss function analysis for binary classification. In: 2010 IEEE International Conference on Data Mining Workshops. Piscataway, NJ: IEEE, 2010, 1281–1288
76. Zhao Y P, Sun J G. Recursive reduced least squares support vector regression. Pattern Recognition, 2009, 42(5): 837–842
77. Zhou S S. Sparse LSSVM in primal using Cholesky factorization for large-scale problems. IEEE Trans Neural Netw Learn Syst, 2016, 27(4): 783–795
78. Zhu W X, Song Y Y, Xiao Y Y. Support vector machine classifier with huberized pinball loss. Eng Appl Artif Intell, 2020, 91: 103635