Optimization
A Journal of Mathematical Programming and Operations Research

ISSN: 0233-1934 (Print) 1029-4945 (Online) Journal homepage: www.tandfonline.com/journals/gopt20

A fast smoothing newton method for bilevel hyperparameter optimization for SVC with logistic loss

Yixin Wang & Qingna Li

To cite this article: Yixin Wang & Qingna Li (30 Aug 2024): A fast smoothing newton method for bilevel hyperparameter optimization for SVC with logistic loss, Optimization, DOI: 10.1080/02331934.2024.2394612

To link to this article: https://doi.org/10.1080/02331934.2024.2394612

OPTIMIZATION
https://doi.org/10.1080/02331934.2024.2394612

Taylor & Francis
Taylor & Francis Group

A fast smoothing newton method for bilevel hyperparameter optimization for SVC with logistic loss

Yixin Wang[a] and Qingna Li[a,b]

[a]School of Mathematics and Statistics, Beijing Institute of Technology, Beijing, People's Republic of China; [b]Beijing Key Laboratory on MCAACI/Key Laboratory of Mathematical Theory and Computation in Information Security, Beijing Institute of Technology, Beijing, People's Republic of China

ABSTRACT

Support vector classification (SVC) with logistic loss has excellent theoretical properties in classification problems where the label values are not continuous. In this paper, we reformulate the hyperparameter selection for SVC with logistic loss as a bilevel optimization problem in which the upper-level problem and the lower-level problem are both based on logistic loss. The resulting bilevel optimization model is converted to a single-level nonlinear programming (NLP) based on the KKT conditions of the lower-level problem. Such NLP contains a set of nonlinear equality constraints and a simple lower-bound constraint. The second-order sufficient condition is characterized, which guarantees that the strict local optimizers are obtained. To solve such NLP, we apply the smoothing Newton method proposed in [Liang L, Sun D., Toh KC. A squared smoothing Newton method for semidefinite programming, 2023] to solve the KKT conditions, which contain one pair of complementarity constraints. We show that the smoothing Newton method has a superlinear convergence rate. Extensive numerical results verify the efficiency of the proposed approach and strict local minimizers can be achieved both numerically and theoretically. In particular, compared with other methods, our algorithm can achieve competitive results while consuming less time than other methods.

ARTICLE HISTORY
Received 17 August 2023
Accepted 14 August 2024

KEYWORDS
Support vector classification with logistic loss; hyperparameter selection; bilevel optimization; smoothing newton method; superlinear convergence

1. Introduction

Support vector classification (SVC) is an optimal margin-based classification technique, which is a classical and well-performed learning method for classification problems in machine learning [1–6]. In those models, hyperparameter selection is a critical issue and has been addressed by many researchers both theoretically and practically [7–10]. Cross validation is usually conducted in order to choose hyperparameters. A classical approach for cross validation is the grid search method [11], where one needs to define a hyperparameter grid, and

CONTACT Qingna Li qnl@bit.edu.cn

search for the optimal hyperparameters to minimize the cross validation error (CV error). Bennett et al. [12] emphasize that one of the drawbacks of the grid search approach is that the continuity of the hyperparameter is ignored by the discretization.

In terms of selecting hyperparameters through bilevel optimization, which is a very active field of applied mathematics, different models and approaches have been considered. The main reason is that bilevel optimization problems can serve as a powerful tool for modeling hierarchical decision making processes [13]. This cross validation optimization problem can be formulated as a bilevel program which is significantly more versatile than the commonly used grid search procedure [14]. Colson et al. [15] proceed towards a general formulation of bilevel optimization and then focus on solution approaches with mathematical program with equilibrium constraints (MPEC). Moore et al. [16] propose a nonsmooth bilevel model and design a proximity control approximation algorithm to select hyperparameters with T-fold cross validation. Okuno et al. [17] propose a bilevel optimization model to select the best hyperparameter for a nonsmooth, possibly non-convex, l_p-regularized problem. They present a smoothing-type algorithm with convergence analysis to solve this bilevel optimization model. Li et al. [13] establish a bilevel programming model for support vector classification, transforming it into an MPEC, and using the global relaxing method (GRM) to solve it, which improved the accuracy of classification prediction. In particular, they show that due to the structure of the problem, the Mangasarian–Fromovitz constraint qualification (MFCQ) holds for such MPEC, which guarantees the convergence of GRM to a C-stationary point. In [18], the authors further investigate the same framework with a lower-level loss function replaced by l_2-loss. Interestingly, MPEC-MFCQ also holds for the resulting MPEC. In [19], the authors proposed an efficient LP-Newton global relaxation method to tackle the MPEC derived from the bilevel model in [9], which involves multiple hyperparameters to deal with feature selection. In [20], the authors further characterized the linearly independent constraint qualification for MPEC (LICQ-MPEC) derived from the bilevel model for hyperparameter selection in kernel SVC.

Besides the l_1-loss SVC and the l_2-loss SVC, the logistic-loss SVC is also widely used. The logistic-loss SVC was originally proposed by Nelder and Wedderburn in 1972 [21]. They show that the logistic-loss function constructed based on entropy and conditional probability has better theoretical properties and it uses the maximum likelihood estimation of the probability log function to fit the parameters with the goal of maximizing the probability sense.

Logistic loss has shown strong superiority in classification prediction [22–27]. Compared with l_1-loss function and l_2-loss function, the advantage of the logistic-loss function lies in two aspects. Firstly, the logistic-loss function is smooth, whereas the l_1-loss function and l_2-loss function are not. By introducing extra variables and inequalities, SVC with l_1-loss and l_2-loss functions can be transferred into smooth optimization problems, which lead to

complementarity constraints in the resulting single-level optimization problem. However, it increases the scale of variables and constraints, and the resulting MPEC is much more difficult to solve. In contrast, as we will see later, the logistic-loss SVC in lower-level problems will lead to a nice equality constraint in the resulting single-level NLP. Secondly, the l_1-loss and l_2-loss functions are susceptible to extreme outliers, and small errors near the boundary ($x_i{}^\top w = 0$) are easy to be ignored, which reduces the accuracy of classification. However, the logistic loss function is sensitive near the boundary ($x_i{}^\top w = 0$) and it reduces the effect of outliers [28]. Therefore, the logistic loss can make the model pay more attention to the classification boundary: errors near the boundary line are strengthened, which are less likely to be ignored in the optimization process, and the influence of extreme outliers is weakened, thereby increasing the stability of the model [29].

Due to the above reasons, logistic-loss SVC is extensively studied and is applied to a wide range of biological and socio-technical systems [30–33,51]. Pregibon [34] develops diagnostic measures to aid the analyst in detecting such observations and in quantifying their effect on various aspects of the maximum likelihood fit of a logistic model. Peduzzi et al. [35] perform a Monte Carlo study to evaluate the effect of the number of events per variable (EPV) based on logistic regression analysis. Chaudhuri et al. [36] use regularized logistic loss to establish a privacy protection mechanism, which balances data privacy protection and classification learning performance to a certain extent. Le et al. [37] proposed a fast and accurate approach that utilises a GPU-based extreme gradient boosting machine using squared logistic loss (SqLL) for bankruptcy forecasting.

To summarize, a natural question is that what will happen to the bilevel model for SVC with logistic loss. This motivates the work in our paper. In this paper, we will study the bilevel optimization model based on logistic loss for selecting the hyperparameter in logistic-loss SVC.

The contributions of the paper are as follows. Firstly, we propose a bilevel optimization model based on logistic functions for hyperparameter selection in a binary SVC. Secondly, we convert the bilevel optimization model to an NLP, and investigate the first and secondorder optimality conditions. Thirdly, we apply the squared smoothing Newton method studied in [38] to solve the KKT system of NLP. A superlinear convergence rate is achieved for the proposed method. Finally, we conduct extensive numerical experiments to verify the efficiency of the proposed method.

The paper is organized as follows. In Section 2, based on T-fold cross validation, we introduce a bilevel optimization model with logistic loss functions to select an optimal hyperparameter for SVC. In Section 3, we transfer the bilevel optimization model to an NLP and analyze the interesting properties including critical cone and second-order sufficient conditions. In Section 4, we smooth the KKT conditions of NLP by Huber function, and reformulate the smoothing system as a nonlinear equation system. We apply a squared smoothing Newton method, and prove that the Jacobian matrix of the nonlinear equation system is

nonsingular, which is a key property to guarantee the superlinear convergence of the algorithm. In Section 5, we conduct extensive numerical experiments, and compare the results with other methods. Finally, we conclude the paper in Section 6.

Notation: *We use* $[k]$ *to denote* $\{1, 2, \ldots, k\}$ *for the positive integer k. We use* ':=' *to represent 'define'. For* $v \in \mathbb{R}^m$, *we use* e^{-v} $(\log v)$ *to denote a column vector in* $\mathbb{R}^m$ *with each component* e^{-v_i} $(\log v_i)$, $i \in [m]$. *We use notations in bold to represent a constant matrix or vector with proper size. For example,* $\mathbf{1}_{m\times m}$ *represents an m by m matrix with all elements ones, and* $\mathbf{0}_m$ *represents a column vector of size m with all elements zeros.* I_m *denotes the m by m identity matrix.*

2. Bilevel model for hyperparameter selection in logistic-SVC

In this part, we establish a bilevel model for hyperparameter selection in logistic-loss SVC based on cross validation.

In order to choose the hyperparameter in logistic-loss SVC, the T-fold cross validation works in the following way. First we split the data set evenly into T parts [18], each part denoted by $\{(\bar{x}_i^{(j)}, \bar{y}_i^{(j)})\}_{i=1}^{m_1}, j \in [T]$, where m_1 denotes the size of the j-th part. $\bar{y}_i^{(j)} \in \{\pm 1\}$ is the label corresponding to data $\bar{x}_i^{(j)} \in \mathbb{R}^n, i \in [m_1]$. For each fold $j \in [T]$, we take the j-th part as the validation set, and the rest parts as the training set (denote as $\{(\hat{x}_i^{(j)}, \hat{y}_i^{(j)})\}_{i=1}^{m_2}, j \in [T]$) to train the model. The best parameter is chosen so that the total loss on each validation set is minimized.

To be specified, we start with the lower-level problem (take the j-th lower-level problem as an example), which is to search for a hyperplane ${w^{(j)}}^\top x = 0$ such that the following logistic-loss SVC model is optimized (given $C \geq 0$):

$$\min_{w^{(j)}\in\mathbb{R}^n} \frac{1}{2}\left\|w^{(j)}\right\|_2^2 + C\sum_{i=1}^{m_2} J_{w^{(j)}}\left(\hat{x}_i^{(j)}, \hat{y}_i^{(j)}\right) := F\left(w^{(j)}\right), \quad j \in [T], \tag{1}$$

where $J_{w^{(j)}}(\hat{x}_i^{(j)}, \hat{y}_i^{(j)})$ is the logistic-loss function (for $\hat{y}_i \in \{\pm 1\}$) defined by

$$J_{w^{(j)}}\left(\hat{x}_i^{(j)}, \hat{y}_i^{(j)}\right) = \frac{1}{2}\left(1 + \hat{y}_i^{(j)}\right)\log\left(1 + e^{-{\hat{x}_i^{(j)}}^\top w^{(j)}}\right) + \frac{1}{2}\left(1 - \hat{y}_i^{(j)}\right)\log\left(1 + e^{{\hat{x}_i^{(j)}}^\top w^{(j)}}\right).$$

In terms of the upper-level problem, the aim is to minimize the loss function over the validation set for all the T folds. Here, we also use the logistic loss as our upper-level loss function [39], which is $\log\left(1 + e^{-\bar{y}_i^{(j)}({\bar{x}_i^{(j)}}^\top w^{(j)})}\right)$, $j \in [T]$.

Overall, we reach the following bilevel model to choose the best parameter C:

$$\begin{aligned}
&\min_{C\in\mathbb{R},w^{(j)}\in\mathbb{R}^n,j\in[T]} \frac{1}{Tm_1}\sum_{j=1}^{T}\sum_{i=1}^{m_1}\log\left(1+\mathrm{e}^{-\bar{y}_i^{(j)}\left(\bar{x}_i^{(j)\top}w^{(j)}\right)}\right)\\
&\text{s.t. } C\geq 0,\\
&\text{and for } j=1,2,\ldots,T,\\
&w^{(j)}=\arg\min_{w^{(j)}\in\mathbb{R}^n}\left\{\frac{1}{2}\left\|w^{(j)}\right\|_2^2+C\sum_{i=1}^{m_2}J_{w^{(j)}}\left(\hat{x}_i^{(j)},\hat{y}_i^{(j)}\right)\right\}.
\end{aligned}\tag{2}$$

3. Single level reformulation and the theoretical properties

In this section, we first reformulate the bilevel optimization problem as a single-level optimization problem based on the KKT conditions of the lower-level problems. Then we present some theoretical properties of the single-level problem, including the KKT conditions and the second-order sufficient conditions.

3.1. Single-level reformulation

Note that each lower-level problem is an unconstrained optimization problem with smooth objective function $F(w^{(j)})$. Due to the convexity of the logistic-loss function, the objective function in the lower-level problem is strongly convex. The solution is unique for each lower-level problem (given $C\geq 0$). A natural way to represent the lower-level problem is by its firstorder optimality condition. That is,

$$\nabla F(w^{(j)})=\mathbf{0},\quad j\in[T],$$

which is

$$\begin{aligned}
&w^{(j)}+\frac{1}{2}C\sum_{i=1}^{m_2}\left(h_{\hat{x}_i^{(j)}}\left(w^{(j)}\right)-\hat{y}_i^{(j)}\right)\hat{x}_i^{(j)}=\mathbf{0},\quad h_{\hat{x}_i^{(j)}}\left(w^{(j)}\right)\\
&:=\frac{1-\mathrm{e}^{-\hat{x}_i^{(j)\top}w^{(j)}}}{1+\mathrm{e}^{-\hat{x}_i^{(j)\top}w^{(j)}}},\ j\in[T].
\end{aligned}\tag{3}$$

By replacing each lower-level problem by (3), the bilevel problem (2) can be equivalently written into the following single-level optimization problem:

$$\begin{aligned} &\min_{C\in\mathbb{R},w^{(j)}\in\mathbb{R}^n,j\in[T]} \frac{1}{Tm_1}\sum_{j=1}^{T}\sum_{i=1}^{m_1}\log\left(1+\mathrm{e}^{-\bar{y}_i^{(j)}\left(\bar{x}_i^{(j)\top}w^{(j)}\right)}\right)\\ &\text{s.t. } C\geq 0,\\ &w^{(j)}+\frac{1}{2}C\sum_{i=1}^{m_2}\left(\hat{x}_i^{(j)}\left(h_{\hat{x}_i^{(j)}}\left(w^{(j)}\right)-\hat{y}_i^{(j)}\right)\right)=\mathbf{0},\quad j\in[T]. \end{aligned} \tag{4}$$

For (4), it is a typical NLP with one simple lower bound constraint and a set of nonlinear equality constraints. Compared with the MPEC reformulation of the bilevel model for hyperparameter selection in l_1-SVC [13] and l_2-SVC [18], the above resulting NLP demonstrates the advantages in the following sense. Firstly, there is no complementarity constraints in (4). Secondly, the objective function and the constraints in NLP are both smooth. Thirdly, the number of the equality constraints in (4) is Tn, which is much smaller than that in MPEC reformulation in [13, 18]. Finally, to solve (4), lots of efficient numerical algorithms for NLP can be applied, whereas for solving MPEC in [13, 18], efficient yet fast algorithms are still highly in need.

(4) can be equivalently written in the matrix form as follows:

$$\begin{aligned} &\min_{C\in\mathbb{R},w\in\mathbb{R}^{Tn}} \frac{1}{Tm_1}\mathbf{1}^\top\log\left(\mathbf{1}+\mathrm{e}^{-Aw}\right)\\ &\text{s.t.}\begin{cases} C\geq 0,\\ w+\dfrac{1}{2}CX\left(h_X(w)-\hat{y}\right)=\mathbf{0}. \end{cases} \end{aligned} \tag{5}$$

Here $w\in\mathbb{R}^{Tn}$, $\hat{y}\in\mathbb{R}^{Tm_2}$, $X\in\mathbb{R}^{Tn\times Tm_2}$, $A\in\mathbb{R}^{Tm_1\times Tn}$, $h_X(w):\mathbb{R}^{Tn}\to\mathbb{R}^{Tm_2}$ are defined by

$$w:=\begin{bmatrix} w^{(1)}\\ \vdots\\ w^{(T)}\end{bmatrix},\quad \hat{y}:=\begin{bmatrix}\hat{y}^{(1)}\\ \vdots\\ \hat{y}^{(T)}\end{bmatrix},\quad X:=\begin{bmatrix} X^{(1)} & \cdots & \mathbf{0}\\ \vdots & \ddots & \vdots\\ \mathbf{0} & \cdots & X^{(T)}\end{bmatrix},$$

$$A:=\begin{bmatrix} A^{(1)} & \cdots & \mathbf{0}\\ \vdots & \ddots & \vdots\\ \mathbf{0} & \cdots & A^{(T)}\end{bmatrix},$$

$$h_X(w):=\begin{bmatrix} h_{X^{(1)}}(w^{(1)})\\ \vdots\\ h_{X^{(T)}}(w^{(T)})\end{bmatrix},\quad \hat{y}^{(j)}:=\begin{bmatrix}\hat{y}_1^{(j)}\\ \vdots\\ \hat{y}_{m_2}^{(j)}\end{bmatrix}\in\mathbb{R}^{m_2},$$

$$X^{(j)} := \left[\hat{x}_1^{(j)}, \ldots, \hat{x}_{m_2}^{(j)}\right] \in \mathbb{R}^{n \times m_2},$$

$$A^{(j)} := \begin{bmatrix} A_1^{(j)} \\ \vdots \\ A_{m_1}^{(j)} \end{bmatrix} \in \mathbb{R}^{m_1 \times n}, \quad A_i^{(j)} := \bar{y}_i^{(j)} \bar{x}_i^{(j)^\top},$$

$$h_{X^{(j)}}(w^{(j)}) := \begin{bmatrix} h_{\hat{x}_1^{(j)}}(w^{(j)}) \\ h_{\hat{x}_2^{(j)}}(w^{(j)}) \\ \vdots \\ h_{\hat{x}_{m_2}^{(j)}}(w^{(j)}) \end{bmatrix} \in \mathbb{R}^{m_2}, \quad j \in [T].$$

3.2. KKT conditions for (5)

The Lagrange function of problem (5) is as follows:

$$\begin{aligned} L(C, w; \lambda, \mu) &= \frac{1}{Tm_1} \mathbf{1}^\top \log\left(\mathbf{1} + \mathrm{e}^{-Aw}\right) - \lambda C + \mu^\top \left(w + \frac{1}{2} CX\left(h_X(w) - \hat{y}\right)\right) \\ &= \frac{1}{Tm_1} f(w) - \lambda C + \mu^\top \left(w + Cg(w)\right), \end{aligned} \tag{6}$$

where $\mu \in \mathbb{R}^{Tn}$, $\lambda \in \mathbb{R}$ are the Lagrange multipliers corresponding to equality and inequality constraints, respectively, and

$$f(w) := \mathbf{1}^\top \log\left(\mathbf{1} + \mathrm{e}^{-Aw}\right), \quad g(w) := \frac{1}{2} X\left(h_X(w) - \hat{y}\right).$$

The KKT conditions of (5) are given by

$$\nabla_C L(C, w; \lambda, \mu) = -\lambda + \mu^\top g(w) = 0, \tag{7a}$$

$$\nabla_w L(C, w; \lambda, \mu) = \frac{1}{Tm_1} \nabla f(w) + \left(I_{Tn} + C \nabla g(w)\right) \mu = \mathbf{0}, \tag{7b}$$

$$0 \leq \lambda \perp C \geq 0, \tag{7c}$$

$$w + Cg(w) = \mathbf{0}, \tag{7d}$$

with

$$\nabla f(w) = -A^\top s(w) \in \mathbb{R}^{Tn}, \quad \nabla g(w) = X \mathrm{Diag}(u(w)) X^\top \in \mathbb{R}^{Tn \times Tn}, \tag{8}$$

$$s(w) = \begin{bmatrix} s^{(1)}(w^{(1)}) \\ \vdots \\ s^{(T)}(w^{(T)}) \end{bmatrix} \in \mathbb{R}^{Tm_1},$$

$$s^{(j)}(w^{(j)}) = \begin{bmatrix} \left(1 + \mathrm{e}^{A_1^{(j)} w^{(j)}}\right)^{-1} \\ \vdots \\ \left(1 + \mathrm{e}^{A_{m_1}^{(j)} w^{(j)}}\right)^{-1} \end{bmatrix} \in \mathbb{R}^{m_1}, \quad j \in [T], \tag{9}$$

$$u(w) = \begin{bmatrix} u^{(1)}(w^{(1)}) \\ \vdots \\ u^{(T)}(w^{(T)}) \end{bmatrix} \in \mathbb{R}^{Tm_2},$$

$$u^{(j)}(w^{(j)}) = \begin{bmatrix} \left(2 + \mathrm{e}^{-\hat{x}_1^{(j)\top} w^{(j)}} + \mathrm{e}^{\hat{x}_1^{(j)\top} w^{(j)}}\right)^{-1} \\ \vdots \\ \left(2 + \mathrm{e}^{-\hat{x}_{m_2}^{(j)\top} w^{(j)}} + \mathrm{e}^{\hat{x}_{m_2}^{(j)\top} w^{(j)}}\right)^{-1} \end{bmatrix} \in \mathbb{R}^{m_2}. \tag{10}$$

Let $v = [C; w] \in \mathbb{R}^{Tn+1}$. Denote the active index set for inequality constraint $C \geq 0$ as follows:

$$\mathcal{I}(v) = \{1\} \quad \text{if } C = 0 \quad \text{and} \quad \mathcal{I}(v) = \emptyset \text{ if } C > 0. \tag{11}$$

Below we show that for each feasible point $v \in \mathbb{R}^{Tn+1}$ of (5), the linearly independent constraint qualification (LICQ) holds.

Proposition 3.1: *LICQ holds at each feasible point $v \in \mathbb{R}^{Tn+1}$ of (5).*

Proof: For simplicity, denote

$$\theta(v) := w + \frac{1}{2} CX\left(h_X(w) - \hat{y}\right) = \begin{bmatrix} \theta_1(v) \\ \vdots \\ \theta_{Tn}(v) \end{bmatrix} \in \mathbb{R}^{Tn}.$$

LICQ holds at v if and only if the following vectors:

$$\{\nabla\theta_i(v) \mid i \in [Tn]\} \cup \left\{[1, 0, 0, \ldots, 0]^\top \in \mathbb{R}^{Tn+1} \mid \text{if } \mathcal{I}(v) = \{1\}\right\} \tag{12}$$

are linearly independent.

Note that

$$\begin{aligned} \begin{bmatrix} \nabla\theta_1(v)^\top \\ \vdots \\ \nabla\theta_{Tn}(v)^\top \end{bmatrix} &= \begin{bmatrix} \nabla_C\theta_1(v) & \nabla_w\theta_1(v)^\top \\ \vdots & \vdots \\ \nabla_C\theta_{Tn}(v) & \nabla_w\theta_{Tn}(v)^\top \end{bmatrix} \\ &= \left[g(w) \quad I_{Tn} + CX\mathrm{Diag}(u(w))X^\top\right] \\ &:= \left[g(w) \quad P(v)\right]. \end{aligned} \tag{13}$$

By definition of $u(w)$ in (10), all elements in $u(w)$ is positive. It implies that $X\text{Diag}(u)X^\top$ is positive semidefinite. Therefore, $P(v)$ is positive definite for any $C \geq 0$.

Consider the following two cases. If $C = 0, \mathcal{I}(v) = \{1\}$. The row vectors in (12) reduces to the row vectors in the following matrix:

$$\begin{bmatrix} g(w) & P(v) \\ 1 & \mathbf{0}_{1\times Tn} \end{bmatrix} = \begin{bmatrix} g(w) & I_{Tn} \\ 1 & \mathbf{0}_{1\times Tn} \end{bmatrix} \in \mathbb{R}^{(Tn+1)\times(Tn+1)}.$$

Obviously, the row vectors in the above matrix are linearly independent, implying LICQ holds. If $C > 0, \mathcal{I}(v) = \emptyset$. The row vectors in (13) are linearly independent due to the positive definiteness of $P(v)$. Overall, LICQ holds at any feasible point v of (5). ■

3.3. Second-order optimality conditions for (5)

To further analyze the second-order optimality conditions for (5), we first need to characterize the critical cone proposed in [40] (Chapter 12, Page 342). Assume that (v^*, λ^*, μ^*) satisfies the KKT system (7).

Proposition 3.2: *Let (v^*, λ^*, μ^*) satisfy the KKT conditions (7). The critical cone $\mathcal{C}(v^*, \lambda^*, \mu^*)$ takes the following form:*

(i) *If $C^* > 0$, $\mathcal{C}(v^*, \lambda^*, \mu^*) = \{d \in \mathbb{R}^{Tn+1} \mid d = \begin{bmatrix} 1 \\ -P(v^*)^{-1}g(w^*) \end{bmatrix} d^C,\ d^C \in \mathbb{R}\}$.*

(ii) *If $C^* = 0$ and $\lambda^* = 0$, $\mathcal{C}(v^*; \lambda^*, \mu^*) = \{d \mid d = \begin{bmatrix} 1 \\ -g(w^*) \end{bmatrix} d^C,\ d^C \geq 0\}$.*

(iii) *If $C^* = 0$ and $\lambda^* > 0$, $\mathcal{C}(v^*; \lambda^*, \mu^*) = \{\mathbf{0}_{Tn+1}\}$.*

Proof: The set of linearized feasible directions at v^*, denoted by $\mathcal{F}(v^*)$, is given by

$$\begin{aligned} \mathcal{F}(v^*) = \{d = \begin{bmatrix} d^C; d^w \end{bmatrix} \in \mathbb{R}^{Tn+1} \mid d^C \in \mathbb{R}, \quad d^w \in \mathbb{R}^{Tn}, \\ g(w^*)d^C + P(v^*)d^w = 0, \\ d^C \geq 0 \text{ if } \mathcal{I}(v^*) \neq \emptyset\}. \end{aligned}$$

The critical cone at (v^*, λ^*, μ^*) is defined as follows:

$$\begin{aligned} \mathcal{C}(v^*, \lambda^*, \mu^*) &= \{d \in \mathcal{F}(v^*) \mid d^C = 0,\ i \in \mathcal{I}(v^*) \text{ with } \lambda_i^* > 0\} \\ &= \{d = \begin{bmatrix} d^C; d^w \end{bmatrix} \in \mathbb{R}^{Tn+1} \mid d^C \in \mathbb{R},\ d^w \in \mathbb{R}^{Tn},\ g(w^*)d^C \\ &\quad + P(v^*)d^w = 0, \\ &\quad d^C = 0 \text{ if } C^* = 0 \text{ and } \lambda^* > 0, \\ &\quad d^C \geq 0 \text{ if } C^* = 0 \text{ and } \lambda^* = 0\}. \end{aligned} \tag{14}$$

Below we discuss three situations.

Case 1. $C^* > 0$. In this case, $\mathcal{I}(v^*) = \{1\}$. By (14), $\mathcal{C}(v^*, \lambda^*, \mu^*)$ reduces to the following form:

$$\begin{aligned}&\mathcal{C}(v^*, \lambda^*, \mu^*)\\ &= \{d = \left[d^C; d^w\right] \in \mathbb{R}^{Tn+1} \mid d^C \in \mathbb{R},\ d^w \in \mathbb{R}^{Tn},\ g(w^*)d^C + P(v^*)d^w = 0\}.\end{aligned}$$

Note that $P(v^*)$ is positive definite for $C \geq 0$. We obtain that $d^w = -P(v^*)^{-1} g(w^*)d^C$. In this case,

$$\mathcal{C}(v^*, \lambda^*, \mu^*) = \left\{ \begin{bmatrix} 1 \\ -P(v^*)^{-1}g(w^*) \end{bmatrix} d^C \mid d^C \in \mathbb{R} \right\}.$$

Case 2. $C^* = 0$ and $\lambda^* = 0$. In this case, (14) reduces to the following form:

$$\begin{aligned}&\mathcal{C}(v^*, \lambda^*, \mu^*)\\ &= \left\{d = \left[d^C; d^w\right] \in \mathbb{R}^{Tn+1} \mid g(w^*)d^C + d^w = 0,\ d^C \geq 0,\ d^w \in \mathbb{R}^{Tn}\right\}. \quad (15)\end{aligned}$$

One can obtain that $d^w = -g(w)d^C$, which gives that $d \in \mathcal{C}(v^*, \lambda^*, \mu^*)$ takes the following form:

$$d = \begin{bmatrix} 1 \\ -g(w^*) \end{bmatrix} d^C, \quad d^C \geq 0.$$

Case 3. $C^* = 0$ and $\lambda^* > 0$. (14) reduces to the following form:

$$\mathcal{C}(v^*, \lambda^*, \mu^*) = \left\{ d \in \mathbb{R}^{Tn+1} \mid \begin{bmatrix} 1 & \mathbf{0}_{1\times Tn} \\ g(w^*) & I_{Tn} \end{bmatrix} d = \mathbf{0} \right\}.$$

Since $\begin{bmatrix} 1 & \mathbf{0}_{1\times Tn} \\ g(w^*) & I_{Tn} \end{bmatrix}$ is nonsingular, $\mathcal{C}(v^*, \lambda^*, \mu^*) = \{\mathbf{0}_{Tn+1}\}$. The proof is finished. ■

Define $\alpha(\mu, v) \in \mathbb{R}^{Tn\times Tn}$, $z(\mu, v) \in \mathbb{R}$ and $\iota(\mu) \in \mathbb{R}$ by

$$\alpha(\mu, v) := \frac{1}{Tm_1} A^\top \mathrm{Diag}(p(w))A + CX\mathrm{Diag}(X^\top \mu)\mathrm{Diag}(q(w))X^\top, \quad (16a)$$

$$z(\mu, v) := g(w)^\top P(v)^{-1} \left(\alpha(\mu, v) P(v)^{-1} g(w) - -2\nabla g(w)\mu\right), \quad (16b)$$

$$\iota(\mu) := \frac{1}{4}\mu^\top XX^\top X\hat{y} + \frac{1}{16Tm_1}\left(X\hat{y}\right)^\top A^\top AX\hat{y}. \quad (16c)$$

Assumption 3.3: *Assume that* $z(\mu^*, v^*) > 0$.

Assumption 3.4: *Assume that* $\iota(\mu^*) > 0$.

Theorem 3.5: *Let* $v^* = [C^*; w^*] \in \mathbb{R}^{1+Tn}$ *with the Lagrange multiplier* $\gamma^* = [\lambda^*; \mu^*] \in \mathbb{R}^{1+Tn}$ *satisfy the KKT conditions (7). If one of the three following conditions holds, then* v^* *is a strict local minimizer of (5):*

(i) $C^* > 0$ *and Assumption 3.3 holds;*
(ii) $C^* = 0$, $\lambda^* = 0$ *and Assumption 3.4 holds;*
(iii) $C^* = 0$ *and* $\lambda^* > 0$.

Proof: First, we derive the formulation of $\nabla^2_{vv} L(v^*; \lambda^*, \mu^*)$. By the definition of Lagrange function in (6), it holds that

$$\nabla^2_{vv} L(v; \lambda, \mu) = \begin{bmatrix} \nabla^2_{CC} L(v; \lambda, \mu) & \nabla^2_{Cw} L(v; \lambda, \mu) \\ \nabla^2_{Cw} L(v; \lambda, \mu)^\top & \nabla^2_{ww} L(v; \lambda, \mu) \end{bmatrix}.$$

Note that $\nabla^2_{CC} L(v; \lambda, \mu) = 0$, $\nabla^2_{Cw} L(v; \lambda, \mu) = \mu^\top \nabla g(w) = \mu^\top X \mathrm{Diag}(u(w)) X^\top$.

Below we derive $\nabla^2_{ww} L(v; \lambda, \mu)$. By (7b), it holds that

$$\begin{aligned} \nabla^2_{ww} L(v; \lambda, \mu) &= \nabla_w \left(\frac{1}{Tm_1} \nabla f(w) + C \nabla g(w) \mu \right) \quad \text{(by (7b))} \\ &= \nabla_w \left(\frac{-A^\top s(w)}{Tm_1} + CX \mathrm{Diag}(u(w)) X^\top \mu \right) \quad \text{(by (8))} \\ &= \nabla_w \left(\frac{-A^\top s(w)}{Tm_1} + CX \mathrm{Diag}(X^\top \mu) u(w) \right) \\ &:= \frac{1}{Tm_1} A^\top \mathrm{Diag}(p(w)) A + CX \mathrm{Diag}(X^\top \mu) \mathrm{Diag}(q(w)) X^\top, \end{aligned}$$

where

$$p(w) := \begin{bmatrix} p^{(1)}(w^{(1)}) \\ \vdots \\ p^{(T)}(w^{(T)}) \end{bmatrix} \in \mathbb{R}^{Tm_1},$$

$$p^{(j)}(w^{(j)}) := \begin{bmatrix} \frac{e^{A_1^{(j)} w^{(j)}}}{\left(1+e^{A_i^{(j)} w^{(j)}}\right)^2} \\ \vdots \\ \frac{e^{A_{m_1}^{(j)} w^{(j)}}}{\left(1+e^{A_i^{(j)} w^{(j)}}\right)^2} \end{bmatrix} \in \mathbb{R}^{m_1}, \quad j \in [T],$$

$$q(w) := \begin{bmatrix} q^{(1)}(w^{(1)}) \\ \vdots \\ q^{(T)}(w^{(T)}) \end{bmatrix} \in \mathbb{R}^{Tm_2},$$

$$q^{(j)}(w^{(j)}) := \begin{bmatrix} \frac{e^{-\hat{x}_1^{(j)\top} w^{(j)}} \left(e^{-\hat{x}_1^{(j)\top} w^{(j)}} - 1 \right)}{\left(1+e^{-\hat{x}_1^{(j)\top} w^{(j)}} \right)^3} \\ \vdots \\ \frac{e^{-\hat{x}_{m_2}^{(j)\top} w^{(j)}} \left(e^{-\hat{x}_{m_2}^{(j)\top} w^{(j)}} - 1 \right)}{\left(1+e^{-\hat{x}_{m_2}^{(j)\top} w^{(j)}} \right)^3} \end{bmatrix} \in \mathbb{R}^{m_2}.$$

For $d = [d^C; d^w] \in \mathbb{R}^{Tn+1} \in \mathcal{C}(v^*, \lambda^*, \mu^*)$, by Proposition 3.2, the second-order sufficient condition can be discussed in three cases.

(i) $C^* > 0$. For any $d \in \mathcal{C}(v^*, \lambda^*, \mu^*)$, by Proposition 3.2 (i), it holds that

$$\begin{aligned} & d^\top \nabla^2_{vv} L\left(v^*; \lambda^*, \mu^*\right) d \\ &= -2(d^C)^2 {\mu^*}^\top X \mathrm{Diag}(u(w^*)) X^\top P(v^*)^{-1} g(w^*) \\ &\quad + (d^C)^2 \left(P(v^*)^{-1} g(w^*)\right)^\top \nabla^2_{ww} L(v; \lambda^*, \mu^*) P(v^*)^{-1} g(w^*) \\ &= (d^C)^2 z(\mu^*, v^*),\ d^C \in \mathbb{R}. \end{aligned}$$

By Assumption 3.3, $d^\top \nabla^2_{vv} L(v^*; \lambda^*, \mu^*) d > 0$, for any $d \in \mathcal{C}(v^*, \lambda^*, \mu^*),\ d \neq 0$. Then v^* is a strict local minimizer of problem (5).

(ii) $C^* = 0$ and $\lambda^* = 0$. By (7d), we can get $w^* = \mathbf{0}$, and by the definition of $g(w)$, $u(w)$, $p(w)$ and $q(w)$, we get $g(w^*) = -\frac{1}{2} X\hat{y}$, $u(w^*) = p(w^*) = \frac{\mathbf{1}}{4}$ and $q(w^*) = \mathbf{0}$. Then $L(v; \lambda^*, \mu^*) = \frac{1}{4Tm_1} A^\top A$. By Proposition 3.2 (ii), for any $d \in \mathcal{C}(v^*, \lambda^*, \mu^*)$, it holds that

$$\begin{aligned} d^\top \nabla^2_{vv} L\left(v^*; \lambda^*, \mu^*\right) d &= \frac{1}{4}(d^C)^2 {\mu^*}^\top X X^\top X\hat{y} + \frac{1}{16Tm_1}(d^C)^2 \left(X\hat{y}\right)^\top A^\top A X\hat{y} \\ &= (d^C)^2 \iota(\mu^*),\ d^C \geq 0. \end{aligned}$$

By Assumption 3.4, $d^\top \nabla^2_{vv} L(v^*; \lambda^*, \mu^*) d > 0$, for any $d \in \mathcal{C}(v^*, \lambda^*, \mu^*), d \neq 0$. Then v^* is a strict local minimizer of problem (5).

(iii) $C^* = 0$ and $\lambda^* > 0$. Since $\mathcal{C}(v^*, \lambda^*, \mu^*) = \{\mathbf{0}_{Tn+1}\}$, it automatically holds that $d^\top \nabla^2_{vv} L(v^*; \lambda^*, \mu^*) d > 0$. Then v^* is a strict local minimizer of problem (5). ■

4. A squared smoothing newton method

In this section, we will solve the KKT conditions in (7) by a squared smoothing Newton method. First, we smooth the complementarity conditions with Huber smoothing function [38], and then combine the bi-conjugate gradient method

(Bi-CG) [41] and a square smooth Newton method [38] to solve the problem. Finally, we prove the superlinear convergence rate of the proposed method.

By eliminating $\lambda = -\mu^\top g(w)$ by (7a), we get the reduced KKT conditions for problem (5), that is,

$$0 \leq \mu^\top g(w) \perp C \geq 0, \tag{17a}$$

$$\frac{1}{Tm_1}\nabla f(w) + \left(I_{Tn} + C\nabla g(w)\right)\mu = \mathbf{0}, \tag{17b}$$

$$w + Cg(w) = \mathbf{0}. \tag{17c}$$

4.1. Smoothing system based on huber smoothing function

(17a) is a nonsmooth complementarity condition. There are numerous methods to transform this difficult condition to its smoothing equivalent system in the literature. One of the most widely used methods is the smoothing Newton method, which relies on an appropriate smoothing function. In order to deal with the nonsmooth plus function $\max\{0, t\}$, $t \in \mathbb{R}$, the most commonly used smoothing function is Chen-Harker-Kanzow-Smale (CHKS) function, that is,

$$\xi(\epsilon, t) = \frac{\sqrt{t^2 + 4\epsilon^2} + t}{2}, \quad (\epsilon, t) \in \mathbb{R} \times \mathbb{R},$$

which has been extensively studied and used for solving nonlinear complementarity problems [42–44]. Besides, some other smoothing functions whose properties have also been well-studied [45, 46]. However, it is easy to see that $\xi(\epsilon, t)$ maps any negative number t to a positive one when $\epsilon \neq 0$. Thus, it destroys the possible sparsity structure when evaluating the Jacobian of the merit function [38]. Hence, smoothing Newton methods based on the CHKS function would require more computational effort. To resolve this issue, we use the Huber smoothing function proposed by Pinar and Zenios [47] which maps any negative number to zero so that the underlying sparsity structure from the plus function is inherited.

It is easy to verify that the following proposition holds:

$$0 \leq a \perp b \geq 0 \Leftrightarrow b - \max\{b - a, 0\} = 0. \tag{18}$$

By (18), we convert (17a) to its equivalent equation, that is,

$$C - \max\left\{C - \mu^\top g(w), 0\right\} = 0. \tag{19}$$

Then we obtain the following equivalent system of (17)

$$\mathcal{K}(C, w, \mu) = \begin{bmatrix} C - \max\left\{C - \mu^\top g(w), 0\right\} \\ \frac{1}{Tm_1}\nabla f(w) + \left(I_{Tn} + C\nabla g(w)\right)\mu \\ w + Cg(w) \end{bmatrix} = \mathbf{0}. \tag{20}$$

Note that $\rho(t) = \max\{0, t\}, t \in \mathbb{R}$, is not differentiable at $t = 0$. We consider its Huber smoothing (or approximation) function [38] defined as follows:

$$h(\epsilon, t) = \begin{cases} t - \dfrac{|\epsilon|}{2}, & t > |\epsilon| \\ \dfrac{t^2}{2|\epsilon|}, & 0 \le t \le |\epsilon| \\ 0, & t < 0 \end{cases}, \quad \forall\ (\epsilon, t) \in \mathbb{R}\backslash\{0\} \times \mathbb{R},\ h(0, t) = \rho(t). \tag{21}$$

We replace the nonsmooth Equation (19) by the following smooth equation:

$$C - h\left(\epsilon, C - \mu^\top g(w)\right) = 0.$$

Following the idea in [38], we solve the following smoothing system for KKT system (20):

$$\begin{aligned} \hat{\mathcal{E}}(\epsilon, C, \mu, w) &= \begin{bmatrix} \epsilon \\ (1 + \kappa|\epsilon|) C - h\left(\epsilon, C - \mu^\top g(w)\right) \\ \frac{1}{Tm_1} \nabla f(w) + \left(I_{Tn} + C\nabla g(w)\right) \mu \\ w + Cg(w) \end{bmatrix} \\ &:= \begin{bmatrix} \epsilon \\ \mathcal{E}(\epsilon, C, \mu, w) \end{bmatrix} = \mathbf{0}, \end{aligned} \tag{22}$$

where $\kappa > 0$ is a given constant.

Note that $\mathcal{E}(\cdot)$ is a continuously differentiable system around any (ϵ, C, μ, w) for any $\epsilon > 0$. Also, as Liang et al. proved in [38], that it satisfies

$$\mathcal{E}\left(\epsilon^k, C^k, \mu^k, w^k\right) \to \mathcal{K}(C, \mu, w), \text{ as } \left(\epsilon^k, C^k, \mu^k, w^k\right) \to (0, C, \mu, w).$$

Note that adding the perturbation terms $\kappa|\epsilon|C$ for constructing the smoothing function of $\mathcal{K}$ is crucial in our algorithm since it ensures the correctness of our proposed algorithm.

4.2. A squared smoothing newton method

In this subsection, we will solve the nonlinear equation system in (22) by combining the bi-conjugate gradient method (Bi-CG) [41] and a square smooth Newton method proposed by Liang et al. [38], which can greatly improve the training speed. The details of our algorithm is shown in Algorithm 1, where $J_{\hat{\mathcal{E}}}(\epsilon, C, \mu, w)$ is the Jacobian matrix of the smoothing system $\hat{\mathcal{E}}(\cdot)$ in (22).

In the numerical experiments, we can obtain the reduced form (28) by eliminating variable $\Delta\epsilon^k$ in Equations (23) by (26), that is, we apply Bi-CG to solve

Algorithm 1 A squared smoothing Newton method (SN)

Require: $\hat{\epsilon} \in (0,\infty)$, $r \in (0,1)$, $\hat{\eta} \in (0,\infty)$ st. $\delta := \sqrt{2}\max\left\{r\hat{\epsilon}, \hat{\eta}\right\} < 1, \hat{r} \in (0,\infty)$, $\rho \in (0,1)$, $\sigma \in \left(0, \frac{1}{2}\right)$, $\tau \in (0,1]$, $\left(\epsilon^0, C^0, w^0, \mu^0\right) \in \mathbb{R} \times \mathbb{R} \times \mathbb{R}^{Tn} \times \mathbb{R}^{Tn}$.

1: **for** $k \geq 0$ **do**

2: **if** $\hat{\mathcal{E}}\left(\epsilon^k, C^k, \mu^k, w^k\right) = \mathbf{0}$ **then**

3: Output: $\left(\epsilon^k, C^k, \mu^k, w^k\right)$.

4: **else**

5: Compute $\eta_k := \eta\left(\epsilon^k, C^k, \mu^k, w^k\right)$ and $\zeta_k := \zeta\left(\epsilon^k, C^k, \mu^k, w^k\right)$, where

$$\eta\left(\epsilon^k, C^k, \mu^k, w^k\right) = \min\left\{1, \hat{r}\|\hat{\mathcal{E}}\left(\epsilon^k, C^k, \mu^k, w^k\right)\|^\tau\right\},$$
$$\zeta\left(\epsilon^k, C^k, \mu^k, w^k\right) = r\min\left\{1, \|\hat{\mathcal{E}}\left(\epsilon^k, C^k, \mu^k, w^k\right)\|^{1+\tau}\right\}.$$

6: Solve the equation system by bi-conjugate gradient method (Bi-CG):

$$\hat{\mathcal{E}}\left(\epsilon^k, C^k, \mu^k, w^k\right) + J_{\hat{\mathcal{E}}}\left(\epsilon^k, C^k, \mu^k, w^k\right)\begin{bmatrix}\Delta\epsilon^k \\ \Delta C^k \\ \Delta\mu^k \\ \Delta w^k\end{bmatrix} = \begin{bmatrix}\zeta_k\hat{\epsilon} \\ 0 \\ \mathbf{0} \\ \mathbf{0}\end{bmatrix}, \tag{23}$$

approximately such that

$$\|\mathcal{R}_k\| \leq \eta_k\|\mathcal{E}\left(\epsilon^k, C^k, \mu^k, w^k\right) + \nabla_{\epsilon^k}\hat{\mathcal{E}}\left(\epsilon^k, C^k, \mu^k, w^k\right)\|,$$
$$\|\mathcal{R}_k\| \leq \hat{\eta}\|\hat{\mathcal{E}}\left(\epsilon^k, C^k, \mu^k, w^k\right)\|,$$

where

$$\Delta\epsilon^k := -\epsilon^k + \zeta_k\hat{\epsilon}, \tag{24}$$
$$\mathcal{R}_k := \mathcal{E}\left(\epsilon^k, C^k, \mu^k, w^k\right) + J_{\hat{\mathcal{E}}}\left(\epsilon^k, C^k, \mu^k, w^k\right)\begin{bmatrix}\Delta\epsilon^k \\ \Delta C^k \\ \Delta\mu^k \\ \Delta w^k\end{bmatrix}.$$

7: Compute ℓ_k as the smallest non-negative integer ℓ satisfying

$$\begin{aligned}&\psi\left(\epsilon^k + \rho^\ell\Delta\epsilon^k, C^k + \rho^\ell\Delta C^k, \mu^k + \mu^\ell\Delta\mu^k, w^k + \rho^\ell\Delta w^k\right)\\ &\leq \left[1 - 2\sigma\left(1-\delta\right)\rho^\ell\right]\psi\left(\epsilon^k, C^k, \mu^k, w^k\right),\end{aligned} \tag{25}$$

where $\psi\left(\epsilon, C, \mu, w\right) = \|\hat{\mathcal{E}}\left(\epsilon, C, \mu, w\right)\|^2$.

8: Compute

$$amp;\left(\epsilon^{k+1}, C^{k+1}, \mu^{k+1}, w^{k+1}\right)$$
$$amp;= \left(\epsilon^k, C^k, \mu^k, w^k\right) + \rho^{\ell_k}\left(\Delta\epsilon^k, \Delta C^k, \Delta\mu^k, \Delta w^k\right).$$

9: **end if**

10: **end for**

the following system:

$$\begin{bmatrix}1 + \kappa\epsilon^k - h_2 & h_2 g(w^k)^\top & h_2{\mu^k}^\top\nabla g(w^k) \\ \nabla g(w^k)\mu^k & I_{Tn} + C^k\nabla g(w^k) & \nabla_{w^k}\left(\frac{1}{Tm_1}\nabla f(w^k) + C^k\nabla g(w^k)\mu^k\right) \\ g\left(w^k\right) & g\left(w^k\right) & I_{Tn} + C^k\nabla g(w^k)\end{bmatrix}$$

$$\begin{bmatrix} \Delta C^k \\ \Delta \mu^k \\ \Delta w^k \end{bmatrix} = -\begin{bmatrix} \left(1+\kappa\left|\epsilon^k\right|\right) C^k - h\left(\epsilon^k, C^k - {\mu^k}^\top g\left(w^k\right)\right) + (\kappa C^k - h_1)(\zeta_k\hat{\epsilon} - \epsilon^k) \\ \frac{1}{Tm_1}\nabla f(w^k) + \left(I_{Tn} + C^k \nabla g(w^k)\right)\mu^k \\ w^k + C^k g\left(w^k\right) \end{bmatrix}. \tag{28}$$

4.3. Global convergence

In this part, we will investigate the convergence of Algorithm 1 (SN). First, we prove that $J_{\hat{\mathcal{E}}}(\epsilon, C, \mu, w)$ is nonsingular under proper assumptions. Define $v(\mu, v, \epsilon) \in \mathbb{R}$ by

$$v(\mu, v, \epsilon) := 1 + \kappa\epsilon - h_2 + h_2 g\left(w\right)^\top P(v)^{-1}\left(\alpha\left(\mu, v\right) P\left(v\right)^{-1} g\left(w\right) - 2\nabla g\left(w\right)\mu\right).$$

Lemma 4.1: *$J_{\hat{\mathcal{E}}}(\epsilon, C, \mu, w)$ is nonsingular if $\epsilon > 0$ and $z(\mu, v) > -\kappa\epsilon$, where $\kappa > 0$ is a given constant.*

Proof: From (22), note that

$$J_{\hat{\mathcal{E}}}\left(\epsilon, C, \mu, w\right) = \begin{bmatrix} 1 & 0 & \mathbf{0}_{1\times Tn} & \mathbf{0}_{1\times Tn} \\ \kappa C - h_1 & 1+\kappa\epsilon - h_2 & h_2 g(w)^\top & h_2 \mu^\top \nabla g(w) \\ \mathbf{0}_{Tn} & \nabla g(w)\mu & I_{Tn} + C\nabla g(w) & \nabla_w\left(\frac{1}{Tm_1}\nabla f(w) + C\nabla g(w)\mu\right) \\ \mathbf{0}_{Tn} & g\left(w\right) & \mathbf{0}_{Tn\times Tn} & I_{Tn} + C\nabla g(w) \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & \mathbf{0}_{1\times Tn} & \mathbf{0}_{1\times Tn} \\ \kappa C - h_1 & 1+\kappa\epsilon - h_2 & h_2 g(w)^\top & h_2 \mu^\top \nabla g(w) \\ \mathbf{0}_{Tn} & \nabla g(w)\mu & P(v) & \alpha(\mu, v) \\ \mathbf{0}_{Tn} & g\left(w\right) & \mathbf{0}_{Tn\times Tn} & P(v) \end{bmatrix}, \tag{29}$$

where $h_1 = \nabla_\epsilon h(\epsilon, t)$, $h_2 = \nabla_t h(\epsilon, t) \in (0, 1]$, $\epsilon > 0$, $t = C - \mu^\top g(w)$.

To show that $J_{\hat{\mathcal{E}}}(\epsilon, C, \mu, w)$ is nonsingular, it suffices to show that the following system of linear equations:

$$J_{\hat{\mathcal{E}}}\left(\epsilon, C, \mu, w\right)\begin{bmatrix} \Delta\epsilon \\ \Delta C \\ \Delta\mu \\ \Delta w \end{bmatrix} = \mathbf{0} \tag{30}$$

only admits zero solution. It is obvious that $\Delta\epsilon = 0$. (30) reduces to the following equations:

$$\begin{bmatrix} 1+\kappa\epsilon-h_2 & h_2 g(w)^\top & h_2\mu^\top\nabla g(w) \\ \nabla g(w)\mu & P(v) & \alpha(\mu,v) \\ g(w) & \mathbf{0}_{Tn\times Tn} & P(v) \end{bmatrix} \begin{bmatrix} \Delta C \\ \Delta\mu \\ \Delta w \end{bmatrix} = \mathbf{0}. \tag{31}$$

We can solve the equation system (31) as follows. First, note that $P(v)$ is positive definite, then

$$\Delta w = -\Delta C P(v)^{-1} g(w). \tag{32}$$

By eliminating Δw with (32), the equation system (31) is reduced as

$$\begin{bmatrix} 1+\kappa\epsilon-h_2 & h_2 g(w)^\top \\ \nabla g(w)\mu & P(v) \end{bmatrix} \begin{bmatrix} \Delta C \\ \Delta\mu \end{bmatrix} = \Delta C \begin{bmatrix} h_2\mu^\top\nabla g(w) \\ \alpha(\mu,v) \end{bmatrix} P(v)^{-1} g(w). \tag{33}$$

which gives

$$\Delta\mu = \Delta C P(v)^{-1} \left(\alpha(\mu,v) P(v)^{-1} g(w) - \nabla g(w)\mu\right). \tag{34}$$

Substituting $\Delta\mu$ into the first equation of (33) gives

$$\begin{aligned} &\left(1+\kappa\epsilon-h_2+h_2 g(w)^\top P(v)^{-1}\left(\alpha(\mu,v) P(v)^{-1} g(w) - \nabla g(w)\mu\right)\right)\Delta C \\ &\quad = h_2\mu^\top\nabla g(w) P(v)^{-1} g(w)\Delta C, \end{aligned}$$

that is,

$$\begin{aligned} &\left(1+\kappa\epsilon-h_2+h_2 g(w)^\top P(v)^{-1}\left(\alpha(\mu,v) P(v)^{-1} g(w) - 2\nabla g(w)\mu\right)\right)\Delta C \\ &\quad = \left(1+\kappa\epsilon+h_2\left(z(\mu,v)-1\right)\right)\Delta C = \nu(\mu,v,\epsilon)\Delta C = 0. \end{aligned} \tag{35}$$

By $z(\mu,v) > -\kappa\epsilon$ and $h_2 \in (0,1]$, we can get $h_2(z(\mu,v)-1) > -1-\kappa\epsilon$, and then it holds that $\nu(\mu,v,\epsilon) \neq 0$. The equation system (35) only admits zero solution $\Delta C = 0$, leading to $\Delta w = \Delta\mu = \mathbf{0}$ by (32) and (34). Thus, (30) only admits zero solution. Therefore, $J_{\hat{\mathcal{E}}}(\epsilon, C, \mu, w)$ is nonsingular. The proof is completed. ■

By Lemma 9, Lemma 10, Theorem 11 in [38] and Lemma 4.1 above, we can get the global convergence result as follows.

Theorem 4.2: *SN is a well-defined algorithm and generates an infinite sequence* $\{(\epsilon^k, C^k, \mu^k, w^k)\}$. *Any accumulation point* $(\bar{\epsilon}, \bar{C}, \bar{\mu}, \bar{w})$ *of* $\{(\epsilon^k, C^k, \mu^k, w^k)\}$ *with* $z(\bar{\mu}, \bar{v}) > 0$ *is a solution of* $\hat{\mathcal{E}}(\epsilon, C, \mu, w) = \mathbf{0}$.

Proof: Lemma 9, Lemma 10 in [38] also hold in this paper since the termination condition of the SN is same as the algorithm proposed in [38], which implies SN is well-defined and generates an infinite sequence.

$z(\mu, v)$ is continuous, which guarantees that for any accumulation point $(\bar{\epsilon}, \bar{C}, \bar{\mu}, \bar{w})$ with $z(\bar{\mu}, \bar{v}) > 0$, there exists an open neighborhood $\mathcal{U}$ such that $z(\mu, v) > -\kappa\epsilon$ holds for any $(\mu, v) \in \mathcal{U}$, where κ and ϵ are both positive. By Lemma 4.1 above, $J_{\hat{\mathcal{E}}}(\epsilon, C, \mu, w)$ is nonsingular. Note that for k sufficiently large, we have that $(\epsilon^k, C^k, \mu^k, w^k)$ belongs to $\mathcal{U}$ with $\epsilon^k > 0$, leading to $J_{\hat{\mathcal{E}}}(\epsilon^k, C^k, \mu^k, w^k)$ is nonsingular.

Following the proof in Theorem 11 in [38], we obtain that $\bar{\psi} = 0$, which implies that $\hat{\mathcal{E}}(\bar{\epsilon}, \bar{C}, \bar{\mu}, \bar{w}) = \mathbf{0}$. The proof is completed. ■

Next, we verified that $\partial\hat{\mathcal{E}}(\epsilon^*, C^*, \mu^*, w^*)$ is nonsingular under a proper assumption, which guarantees that SN admits a superlinear convergence rate under the same assumption.

Proposition 4.3: *Let $(\epsilon^*, C^*, \mu^*, w^*)$ be such that $\hat{\mathcal{E}}(\epsilon, C, \mu, w) = \mathbf{0}$. If $z(\mu^*, v^*) > 0$, then every element in $\partial\hat{\mathcal{E}}(\epsilon^*, C^*, \mu^*, w^*)$ is nonsingular.*

Proof: Let $U \in \partial\hat{\mathcal{E}}(\epsilon^*, C^*, \mu^*, w^*)$. Since $\epsilon^* = 0$, then

$$U\begin{bmatrix}\Delta\epsilon\\ \Delta C\\ \Delta\mu\\ \Delta w\end{bmatrix} = \begin{bmatrix}1 & 0 & \mathbf{0}_{1\times Tn} & \mathbf{0}_{1\times Tn}\\ \kappa C - h_1^* & 1-h_2^* & h_2^* g(w^*)^\top & h_2^* {\mu^*}^\top \nabla g(w^*)\\ \mathbf{0}_{Tn} & \nabla g(w^*)\mu^* & P(v^*) & \alpha(\mu^*, v^*)\\ \mathbf{0}_{Tn} & g(w^*) & \mathbf{0}_{Tn\times Tn} & P(v^*)\end{bmatrix}\begin{bmatrix}\Delta\epsilon\\ \Delta C\\ \Delta\mu\\ \Delta w\end{bmatrix},$$

where $t^* = C^* - {\mu^*}^\top g(w^*)$, $h_2^* \in \partial_t h(0, t^*) = [0, 1]$,

$$h_1^* \in \partial_\epsilon h(0, t^*) = \begin{cases}\left[-\frac{1}{2}, \frac{1}{2}\right], & t^* > 0,\\ 0, & t^* \le 0.\end{cases}$$

Here, $\partial_t h(0, t^*)$ and $\partial_\epsilon h(0, t^*)$ denote the generalized gradient of $h(\epsilon, t)$ with respect to t and ϵ at $(0, t^*)$ respectively.

To show that U is nonsingular, it suffices to show that the following system of linear equations:

$$U\begin{bmatrix}\Delta\epsilon\\ \Delta C\\ \Delta\mu\\ \Delta w\end{bmatrix} = \mathbf{0} \tag{36}$$

only admits zero solution. It is obvious that $\Delta\epsilon = 0$. (36) reduces to the following equations:

$$\begin{bmatrix}1-h_2^* & h_2^* g(w^*)^\top & h_2^* {\mu^*}^\top \nabla g(w^*)\\ \nabla g(w^*)\mu^* & P(v^*) & \alpha(\mu^*, v^*)\\ g(w^*) & \mathbf{0}_{Tn\times Tn} & P(v^*)\end{bmatrix}\begin{bmatrix}\Delta C\\ \Delta\mu\\ \Delta w\end{bmatrix} = \mathbf{0}. \tag{37}$$

Similar to the proof in Lemma 4.1, by (32) and (34), the equation system (37) can be reduced as

$$\left(1-h_2^*+h_2^* g\left(w^*\right)^{\top} P(v^*)^{-1}\left(\alpha\left(\mu^*, v^*\right) P\left(v^*\right)^{-1} g\left(w^*\right)-2 \nabla g\left(w^*\right) \mu^*\right)\right) \Delta C = (1+h_2^*(z(\mu^*, v^*)-1)) \Delta C = 0. \tag{38}$$

By $z(\mu^*, v^*) > 0$ and $h_2^* \in [0,1]$, we can get $1 + h_2^*(z(\mu^*, v^*) - 1) \neq 0$. The equation system (38) only admits zero solution $\Delta C = 0$, leading to $\Delta\mu = \Delta w = \mathbf{0}$. Thus, (36) only admits zero solution. Hence, $U \in \partial\hat{\mathcal{E}}(\epsilon^*, C^*, \mu^*, w^*)$ is nonsingular. ■

Theorem 4.4: *Let $(\bar{\epsilon}, \bar{C}, \bar{\mu}, \bar{w})$ be an accumulation point of the infinite sequence $\{(\epsilon^k, C^k, \mu^k, w^k)\}$ generated by SN. Suppose that $z(\bar{\mu}, \bar{v}) > 0$. The whole sequence converges to $(\epsilon^*, C^*, \mu^*, w^*)$ superlinearly, i.e.*

$$\left\|\left(\epsilon^{k+1}-\epsilon^*, C^{k+1}-C^*, \mu^{k+1}-\mu^*, w^{k+1}-w^*\right)\right\| = O\left(\left\|\left(\epsilon^k-\epsilon^*, C^k-C^*, \mu^k-\mu^*, w^k-w^*\right)\right\|^{1+\tau}\right),$$

where $\tau \in (0,1]$ is a parameter of SN. Therefore, SN converges superlinearly to its accumulation point $(\epsilon^*, C^*, \mu^*, w^*)$.

Proof: By Theorem 4.2, we see that $\hat{\mathcal{E}}(\bar{\epsilon}, \bar{C}, \bar{\mu}, \bar{w}) = \mathbf{0}$ and in particular, $\bar{\epsilon} = 0$. By Proposition 4.3, every element in $\partial\hat{\mathcal{E}}(0, \bar{C}, \bar{\mu}, \bar{w})$ is nonsingular with $z(\bar{\mu}, \bar{v}) > 0$. By Proposition 3.1 in [48], for all k sufficiently large, it holds that

$$\left\|J_{\hat{\mathcal{E}}}\left(\epsilon^k, C^k, \mu^k, w^k\right)^{-1}\right\| = O(1). \tag{39}$$

Denote $\bar{b} := (\bar{\epsilon}, \bar{C}, \bar{\mu}, \bar{w})^{\top}$, $b^k := (\epsilon^k, C^k, \mu^k, w^k)^{\top}$, $\hat{\mathcal{E}}_* := \hat{\mathcal{E}}(\bar{\epsilon}, \bar{C}, \bar{\mu}, \bar{w})$, $\Delta b^k := (\Delta\epsilon^k, \Delta C^k, \Delta\mu^k, \Delta w^k)^{\top}$, $\hat{\mathcal{E}}_k := \hat{\mathcal{E}}(\epsilon^k, C^k, \mu^k, w^k)$ and $J_k = J_{\hat{\mathcal{E}}}(\epsilon^k, C^k, \mu^k, w^k)$ for $k \geq 0$. Then, we can verify that

$$\begin{aligned}\left\|b^k+\Delta b^k-\bar{b}\right\| &= \left\|b^k+J_k^{-1}\left(\begin{pmatrix}\zeta_k\hat{\epsilon}\\ \mathcal{R}_k\end{pmatrix}-\hat{\mathcal{E}}_k\right)\right\| \\ &= \left\|-J_k^{-1}\left(\hat{\mathcal{E}}_k-J_k(b^k-\bar{b})-\begin{pmatrix}\zeta_k\hat{\epsilon}\\ \mathcal{R}_k\end{pmatrix}\right)\right\| \\ &= O\left(\left\|\hat{\mathcal{E}}_k-J_k(b^k-\bar{b})\right\|\right)+O\left(\left\|\hat{\mathcal{E}}_k\right\|^{1+\tau}\right)+O\left(\|\mathcal{R}_k\|\right).\end{aligned} \tag{40}$$

Since $h(\epsilon, t)$ is strongly semismooth everywhere [38], $\hat{\mathcal{E}}$ is also strongly semismooth at $\bar{b}$. Thus, for k sufficiently large, it holds that

$$\left\|\hat{\mathcal{E}}_k - J_k(b^k - \bar{b})\right\| = O\left(\left\|b^k - \bar{b}\right\|^2\right). \tag{41}$$

Since $\hat{\mathcal{E}}$ is locally Lipschitz continuous at $\bar{b}$, for all k sufficiently large, it holds that

$$\left\|\hat{\mathcal{E}}_k\right\| = \left\|\hat{\mathcal{E}}_k - \hat{\mathcal{E}}_*\right\| = O\left(\left\|b^k - \bar{b}\right\|\right). \tag{42}$$

By Theorem 14 in [38], we can see that

$$\|\mathcal{R}\| \leq O\left(\left\|b^k - \bar{b}\right\|^{1+\tau}\right), \quad \tau \in (0, 1],$$

which together with (40), (41) and (42) implies that

$$\left\|b^k + \Delta b^k - \bar{b}\right\| = O\left(\left\|b^k - \bar{b}\right\|^{1+\tau}\right), \quad \tau \in (0, 1].$$

Therefore, we get for k sufficiently large that

$$\begin{aligned}
&\psi(b^k + \Delta b^k) \\
&= \left\|\hat{\mathcal{E}}(b^k + \Delta b^k)\right\|^2 = O\left(\left\|b^k + \Delta b^k - \bar{b}\right\|^2\right) = O\left(\left\|b^k - \bar{b}\right\|^{2(1+\tau)}\right) \\
&= O\left(\left\|\hat{\mathcal{E}}_k\right\|^{2(1+\tau)}\right) = O\left(\left\|\psi(b^k)\right\|^{1+\tau}\right) = o\left(\left\|\psi(b^k)\right\|\right), \quad \tau \in (0, 1],
\end{aligned}$$

which shows that, for k sufficient large, $b^{k+1} = b^k + \Delta b^k$, i.e. the unit step is eventually accepted with a given constant $\tau \in (0, 1]$. The proof is completed. ■

5. Numerical results

In this section, we present the cross validation algorithm for selecting the hyperparameter C in SVC, as shown in Algorithm 2. All the numerical tests are conducted in Matlab R2018b. All the data sets are collected from the LIBSVM library: https://www.csie.ntu.edu.tw/cjlin/ libsvmtools/datasets/. The data descriptions are shown in Table 1.

Our method is denoted as SN. We use imSN (exSN) to denote the SN method where the Jacobian matrix in (23) is formed implicitly (explicitly). We compare our methods with three other approaches: SNO-CV, where the optimal hyperparameter C is selected by the SNOPT solver [49]; GS-UNC and GS-SNO, in both of which the hyperparameter C is searched in a grid range [9, 50]

$$C \in \{0.5 \times 10^{-4}, 10^{-4}, 0.5 \times 10^{-3}, 10^{-3}, 0.5 \times 10^{-2}, 10^{-2}, 0.5 \times 10^{-1}, 10^{-1},$$

Table 1. Descriptions of data sets.

No.	Data set	p_1	p_2	n	No.	Data set	p_1	p_2	n
1	fourclass	300	562	2	11	a2a	1800	465	119
2	diabetes	540	228	8	12	a3a	1200	985	122
3	breast-cancer	540	143	10	13	a4a	1200	581	122
4	heart	162	108	13	14	a6a	2250	8970	122
5	australian	600	90	14	15	a7a	2700	13400	122
6	svmguide4	30	11	22	16	a9a	3300	29261	123
7	german.number	600	400	24	17	w1a	1500	977	300
8	ionosphere	243	108	34	18	w2a	1800	1670	300
9	sonar	150	58	60	19	w4a	3000	4366	300
10	phishing	1500	500	68	20	w5a	3000	6888	300

Algorithm 2 The cross-validation algorithm (SN-CV)

1: Given T, split the data set into a subset Ω with p_1 points and a hold-out test set Θ with p_2 points. The set Ω is equally partitioned into T pairwise disjoint subsets, one for each fold.

2: **Select** an optimal hyperparameter C by SN in Algorithm 1.

3: **Post − processingprocedure**. The regularization hyperparameter $\hat{C}$ is rescaled by a factor $T/(T-1)$. This gives the final classifier $\hat{w}$.

$$0.5, 1, 0.5 \times 10^1, 10^1, 0.5 \times 10^2, 10^2, 0.5 \times 10^3, 10^3, 0.5 \times 10^4, 10^4\},$$

and the lower-level problems defined as (1) are solved using the given C. In GS-UNC, the optimization problem is solved by the MATLAB function *fminunc*, while in GS-SNO, the same problem is solved by SNOPT solver.

The parameters of the five methods are set as follows. For imSN and exSN, we set the initial values as $\epsilon^0 = 1$, $C^0 = 1$, $\mu^0 = 0$, $w^0 = 0$, $r = \hat{r} = 0.6$, $\hat{\eta} = 0.2$, $\rho = 0.5$, $\sigma = 1 \times 10^{-8}$, $\tau = 0.2$, $\hat{\epsilon} = 0.5$, $\kappa = 1$. For SNO-CV, we set $C^0 = 1$, $\mu^0 = \mathbf{0}$, $w^0 = \mathbf{0}$.

We compare the aforementioned methods in the following three aspects:

(1) CPU time (t), which is the time of hyperparameter optimization by cross validation in different methods with the given data sets in Table 1.

(2) Test error ($E_t(\%)$) as defined by

$$E_t = \frac{1}{p_2} \sum_{(x_i, y_i) \in \Theta} \frac{1}{2} |\text{sign}(x_i^\top w^*) - y_i|,$$

where Θ is the test set.

(3) Cross validation error (E_{CV}) as defined in the objective function of problem (4).

The results are reported in Table 2, where we mark the winners in bold and they are also illustrated in Figures 1–3. We have the following observations.

Table 2. Computational results for $T = 3$.

No.	Data set	(m_{CV}, n_{CV}, γ)	Method	C	t	$E_t(\%)$	E_{CV}
1	fourclass	(7, 6, 14)	imSN	0.860	**0.028**	**22.776**	0.4671
			exSN	0.860	0.055	**22.776**	0.4671
			SNO-CV	4.588	0.152	23.132	**0.4618**
			GS-UNC	0.15	0.094	22.954	0.5132
			GS-SNO	0.15	0.067	22.954	0.5132
2	diabetes	(25, 24, 50)	imSN	0.800	**0.066**	**20.175**	0.5094
			exSN	0.800	0.082	20.175	0.5094
			SNO-CV	4.688	0.795	**20.175**	**0.5025**
			GS-UNC	7.5	0.238	**20.175**	0.5026
			GS-SNO	7.5	0.319	**20.175**	0.5026
3	breast-cancer	(31, 30, 62)	imSN	0.791	**0.115**	**0.000**	0.1272
			exSN	0.791	0.136	**0.000**	0.1272
			SNO-CV	3.528	0.389	**0.000**	**0.1217**
			GS-UNC	7.5	0.321	**0.000**	0.1225
			GS-SNO	7.5	0.379	**0.000**	0.1225
4	heart	(40, 39, 80)	imSN	0.483	**0.023**	**15.741**	**0.4291**
			exSN	0.483	0.031	**15.741**	**0.4291**
			SNO-CV	0.499	0.023	**15.741**	**0.4291**
			GS-UNC	0.075	0.156	16.667	0.4727
			GS-SNO	0.075	0.172	16.667	0.4727
5	australian	(43, 42, 86)	imSN	0.381	**0.239**	**11.111**	0.3564
			exSN	0.381	0.258	**11.111**	0.3564
			SNO-CV	1.287	0.807	14.444	**0.3552**
			GS-UNC	7.5	0.397	14.444	0.3578
			GS-SNO	7.5	0.587	14.444	0.3578
6	svmguide4	(67, 66, 134)	imSN	1.288	**0.005**	**0.000**	0.0653
			exSN	1.288	0.006	**0.000**	0.0653
			SNO-CV	49509	0.337	**0.000**	**0.0000**
			GS-UNC	0.00015	0.131	**0.000**	0.6918
			GS-SNO	0.00015	0.169	**0.000**	0.6918
7	german.number	(73, 72, 146)	imSN	0.226	**0.247**	23.000	0.5094
			exSN	0.246	0.263	23.000	0.5095
			SNO-CV	0.196	0.484	**22.750**	**0.5092**
			GS-UNC	1500	0.719	24.500	0.5292
			GS-SNO	1500	1.152	24.500	0.5292
8	ionosphere	(103, 102, 206)	imSN	0.135	0.101	**0.926**	**0.5980**
			exSN	0.135	**0.073**	**0.926**	**0.5980**
			SNO-CV	0.123	0.110	**0.926**	**0.5980**
			GS-UNC	15	0.459	1.852	1.1415
			GS-SNO	15	0.777	1.852	1.1415
9	sonar	(181, 180, 362)	imSN	0.566	0.116	**27.586**	**0.5333**
			exSN	0.566	**0.092**	**27.586**	**0.5333**
			SNO-CV	0.572	0.124	**27.586**	**0.5333**
			GS-UNC	0.75	1.183	31.034	0.5350
			GS-SNO	0.75	1.853	31.034	0.5350
10	phishing	(205, 204, 410)	imSN	1.672	**0.983**	**5.400**	0.2276
			exSN	1.672	5.437	**5.400**	0.2276
			SNO-CV	96.131	25.584	6.400	**0.1505**
			GS-UNC	15	2.890	6.200	0.1600
			GS-SNO	15	5.511	6.200	0.1600
No.	Data set	(m_{CV}, n_{CV}, γ)	Method	C	t	$E_t(\%)$	E_{CV}
11	a2a	(358, 357, 716)	imSN	0.294	**1.058**	**15.699**	**0.3823**
			exSN	0.278	3.179	15.914	**0.3823**
			SNO-CV	0.303	5.704	**15.699**	**0.3823**
			GS-UNC	75	4.693	17.849	0.4126
			GS-SNO	75	11.684	17.849	0.4126
12	a3a	(367, 366, 734)	imSN	0.466	**0.611**	16.447	0.3719
			exSN	0.421	1.647	**16.345**	0.3721
			SNO-CV	0.611	2.774	**16.345**	**0.3717**
			GS-UNC	0.15	3.445	16.447	0.3795
			GS-SNO	0.15	9.300	16.447	0.3795

(*continued*).

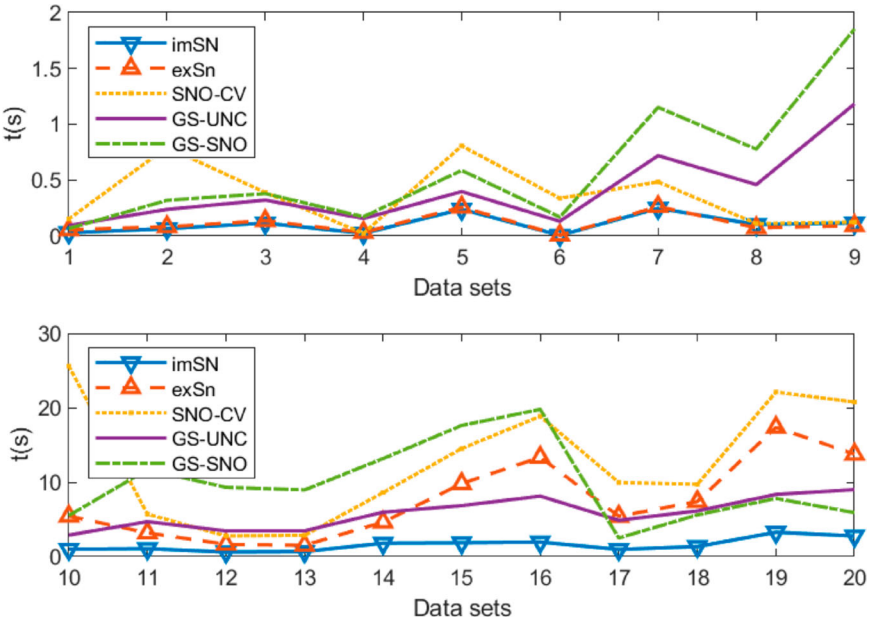

Figure 1. The comparison on CPU time (t).

Firstly, for the CPU time (t) summarized in Figure 1, our imSN method takes the shortest time among the five methods in almost all data sets. In particular, for the data sets with a large number of features (No.10 to No.20 where $n_{CV} \geq 200$), the CPU time of imSN is much less than that of other remaining methods. Besides, in terms of the two SN methods, we can find that there is a difference caused by the number of sample feature. The CPU time of exSN is close to that of imSN for features smaller than about 60, but imSN is much more efficient in large data sets with features more than 60. This reflects the advantages of implicit methods in solving large-scale problems. Secondly, as shown in Figure 2, the methods based on bilevel optimization (i.e. imSN, exSN, SNO-CV) perform better than the two grid search methods (i.e. GS-UNC, GS-SNO) in terms of test error ($E_t(\%)$), implying that our bilevel optimization approaches are more capable of classification prediction. Thirdly, in terms of cross validation error (E_{CV}) in Figure 3, SNO-CV is the best, and the two SN methods are very close to SNO-CV in most data sets. Finally, Figure 4 demonstrates the typical decrease of $\|\hat{\mathcal{E}}\|$ for some data sets, where one can indeed observe the superlinear convergence rate of our algorithm.

Overall, the two SN methods are able to achieve satisfactory results, which are competitive with SNO-CV, and far superior to two GS methods.

Next, we report some further information of the solutions computed by two SN methods in Table 3. (i) The number of iterations in outer loops (SN) and inner loops (BiCG). (ii) $z(\mu^*, \nu^*)$ given in (16b). (iii) The violation of the nonlinear system in terms of $\|\hat{\mathcal{E}}\|_2$, $\|\epsilon\|_2$, and $\|\mathcal{E}\|_2$.

Table 2. Continued.

No.	Data set	(m_{CV}, n_{CV}, γ)	Method	C	t	$E_t(\%)$	E_{CV}
13	a4a	(367, 366, 734)	imSN	0.543	**0.672**	16.448	0.3437
			exSN	0.490	1.518	16.504	0.3440
			SNO-CV	0.747	2.849	**16.392**	**0.3433**
			GS-UNC	0.75	3.464	**16.392**	**0.3433**
			GS-SNO	0.75	8.952	**16.392**	**0.3433**
14	a6a	(367, 366, 734)	imSN	0.427	**1.798**	15.875	0.3552
			exSN	0.364	4.634	**15.753**	0.3554
			SNO-CV	0.544	8.600	15.909	**0.3550**
			GS-UNC	1.5	5.956	16.042	0.3571
			GS-SNO	1.5	13.167	16.054	0.3571
15	a7a	(367, 366, 734)	imSN	0.442	**1.852**	**15.463**	0.3463
			exSN	0.442	9.796	**15.463**	0.3463
			SNO-CV	0.649	14.512	15.493	**0.3460**
			GS-UNC	1.5	6.846	15.627	0.3472
			GS-SNO	1.5	17.617	15.627	0.3472
16	a9a	(370, 369, 740)	imSN	0.298	**1.949**	15.584	0.3300
			exSN	0.356	13.344	15.553	**0.3299**
			SNO-CV	0.359	18.841	**15.546**	**0.3299**
			GS-UNC	0.15	8.121	15.731	0.3327
			GS-SNO	0.15	19.773	15.731	0.3327
17	w1a	(901, 900, 1802)	imSN	0.890	**0.950**	2.354	0.1620
			exSN	0.890	5.410	2.354	0.1620
			SNO-CV	4.153	9.959	**2.252**	**0.1506**
			GS-UNC	0.075	4.877	2.661	0.2290
			GS-SNO	0.075	2.494	2.661	0.2290
18	w2a	(901, 900, 1802)	imSN	0.734	**1.346**	2.635	0.1731
			exSN	0.734	7.401	2.635	0.1731
			SNO-CV	2.602	9.706	**2.575**	**0.1655**
			GS-UNC	0.0015	6.137	2.934	0.5036
			GS-SNO	0.0015	5.602	2.934	0.5036
19	w4a	(901, 900, 1802)	imSN	0.789	**3.265**	**2.176**	0.1621
			exSN	0.789	17.359	**2.176**	0.1621
			SNO-CV	2.303	22.107	2.199	**0.1564**
			GS-UNC	0.75	8.352	**2.176**	0.1627
			GS-SNO	0.75	7.810	**2.176**	0.1627
20	w5a	(901, 900, 1802)	imSN	0.734	**2.762**	2.236	0.1716
			exSN	0.734	13.754	2.236	0.1716
			SNO-CV	1.685	20.758	**2.076**	**0.1685**
			GS-UNC	0.00075	8.990	2.570	0.5239
			GS-SNO	0.00075	5.899	2.570	0.5239

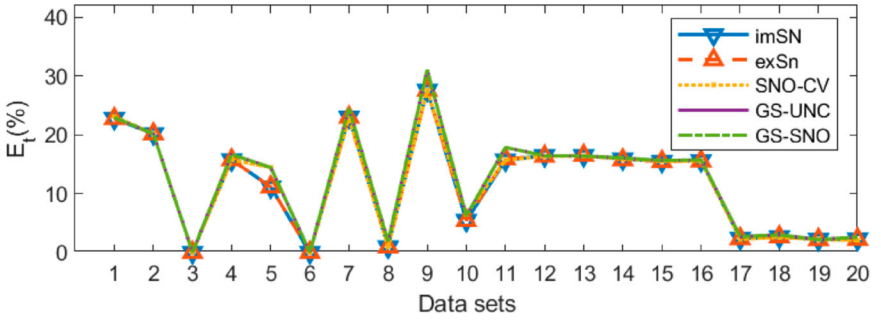

Figure 2. The comparison on test error ($E_t(\%)$).

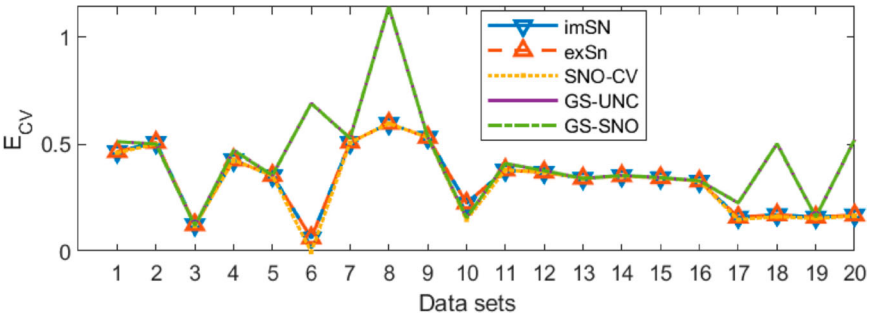

Figure 3. The comparison on cross validation error (E_{CV}).

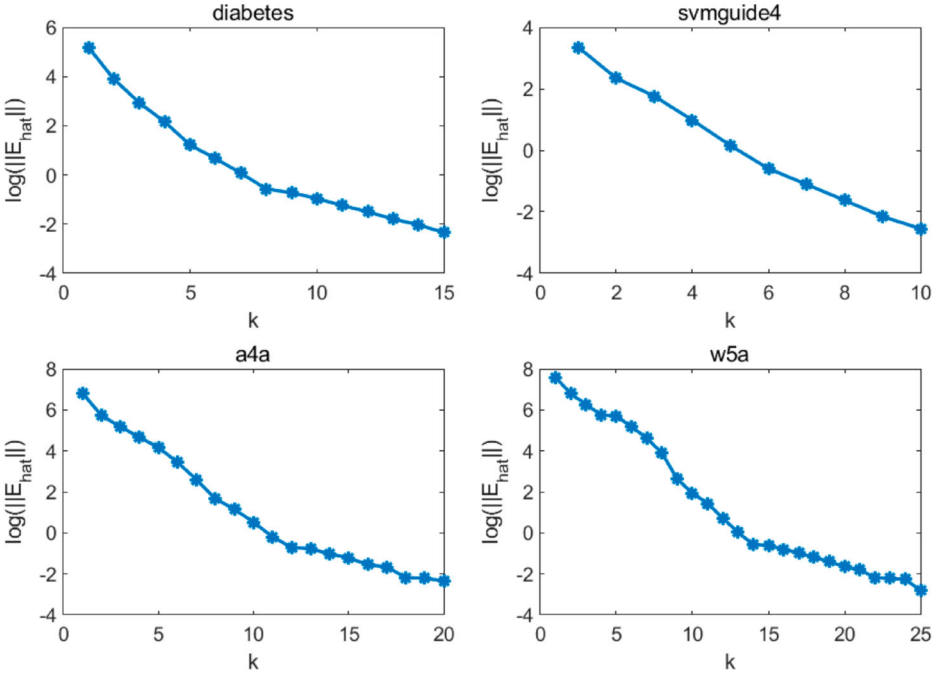

Figure 4. $log\|\hat{\mathcal{E}}\|$ along iterations by imSN.

One can observe from Table 3 that for most data set, the outer loop reaches the termination condition within 30 iterations. Note that all the returned solution C^*s by SN methods are always positive. In terms of $z(\mu^*, v^*)$, it can be seen that $z(\mu^*, v^*)$ is always positive, which satisfies Assumption 3.3 and the condition in Theorems 4.2 and 4.4, implying that SN provides strict local minimizers of (4) and the superlinear convergence rate is achieved. What's more, from the last three columns in Table 3, one can see that the nonlinear system is successfully solved since the returned violation of the system (22) is always smaller than 0.1.

Finally, we compared the hyperparameter selection model for logistic-loss SVC (solved by imSN) with those for l_1-loss SVC [13] (solved by GRCV-l_1[1]) and l_2-loss SVC [18] (solved by GRCV-l_2[2]) in Table 4. The performance of the

Table 3. Further information of SN.

No.	Data set	Method	$(k, iter)$	$z(\mu^*, v^*)$	$\|\hat{\mathcal{E}}\|_2$	$\|\epsilon\|_2$	$\|\mathcal{E}\|_2$
1	fourclass	imSN	(13, 85)	0.068	0.079	0.035	0.071
		exSN	(13, 85)	0.068	0.079	0.035	0.071
2	diabetes	imSN	(14, 153)	0.082	0.096	0.041	0.087
		exSN	(14, 153)	0.082	0.096	0.041	0.087
3	breast-cancer	imSN	(18, 193)	0.064	0.071	0.034	0.063
		exSN	(18, 193)	0.064	0.071	0.034	0.063
4	heart	imSN	(13, 108)	0.187	0.098	0.053	0.082
		exSN	(13, 108)	0.187	0.098	0.053	0.082
5	australian	imSN	(17, 237)	0.100	0.091	0.044	0.080
		exSN	(17, 237)	0.100	0.091	0.044	0.080
6	svmguide4	imSN	(9, 30)	0.103	0.078	0.061	0.049
		exSN	(9, 30)	0.103	0.078	0.061	0.049
7	german.number	imSN	(13, 180)	0.303	0.095	0.051	0.080
		exSN	(14, 206)	0.209	0.078	0.040	0.067
8	ionosphere	imSN	(12, 150)	3.160	0.071	0.043	0.057
		exSN	(12, 150)	3.160	0.071	0.043	0.057
9	sonar	imSN	(12, 202)	0.300	0.091	0.056	0.072
		exSN	(12, 204)	0.300	0.091	0.056	0.072
10	phishing	imSN	(26, 355)	0.063	0.095	0.031	0.090
		exSN	(26, 355)	0.063	0.095	0.031	0.090
11	a2a	imSN	(19, 420)	0.174	0.096	0.050	0.082
		exSN	(19, 392)	0.218	0.088	0.044	0.076
12	a3a	imSN	(19, 351)	0.090	0.095	0.034	0.089
		exSN	(18, 331)	0.123	0.088	0.044	0.076
13	a4a	imSN	(19, 373)	0.087	0.096	0.034	0.090
		exSN	(18, 349)	0.120	0.089	0.044	0.077
14	a6a	imSN	(25, 653)	0.080	0.090	0.035	0.082
		exSN	(19, 436)	0.136	0.091	0.043	0.080
15	a7a	imSN	(23, 625)	0.077	0.099	0.040	0.090
		exSN	(24, 626)	0.078	0.090	0.038	0.082
16	a9a	imSN	(23, 505)	0.194	0.070	0.032	0.062
		exSN	(23, 604)	0.103	0.087	0.037	0.079
17	w1a	imSN	(22, 186)	0.066	0.076	0.032	0.069
		exSN	(22, 186)	0.066	0.076	0.032	0.069
18	w2a	imSN	(23, 235)	0.088	0.096	0.044	0.085
		exSN	(23, 235)	0.088	0.096	0.044	0.085
19	w4a	imSN	(27, 319)	0.071	0.076	0.034	0.068
		exSN	(27, 319)	0.071	0.076	0.034	0.068
20	w5a	imSN	(24, 267)	0.068	0.060	0.025	0.054
		exSN	(24, 267)	0.068	0.060	0.025	0.054

hyperparameter selection model for logistic-loss SVC is the best in classification prediction, which reaches the least error rate ($E_t(\%)$) in most data sets. Moreover, the CPU time for imSN is much shorter than the other two solvers, which verifies the high efficiency of the hyperparameter selection model for logistic-loss SVC and our imSN algorithm.[3]

6. Conclusion

In this paper, we proposed a bilevel optimization model for the hyperparameter selection for SVC in which both the upper-level problem and the lower-level problem are based on the logistic loss. We reformulated the bilevel optimization problem into a single-level NLP based on the KKT condition of the lower-level

Table 4. Computational results of hyperparameter selection for three models.

No.	Data set	Model	E_t(%)	t	No.	Data set	Model	E_t(%)	t
1	fourclass	logistic	22.776	**0.028**	7	german.	logistic	23.000	**0.247**
		l_1-loss	**22.242**	21.022		number	l_1-loss	**22.750**	140.384
		l_2-loss	23.132	6.626			l_2-loss	23.500	424.650
2	diabetes	logistic	**20.175**	**0.066**	8	ionosphere	logistic	**0.926**	**0.101**
		l_1-loss	21.491	48.491			l_1-loss	4.630	7.541
		l_2-loss	**20.175**	54.514			l_2-loss	3.704	21.648
3	breast-	logistic	**0.000**	**0.115**	9	sonar	logistic	27.586	**0.116**
	cancer	l_1-loss	1.399	56.023			l_1-loss	**24.138**	2.985
		l_2-loss	**0.000**	120.402			l_2-loss	25.862	3.246
4	heart	logistic	15.741	**0.023**	10	phishing	logistic	5.400	**0.983**
		l_1-loss	**13.889**	3.099			l_1-loss	**5.000**	1662.824
		l_2-loss	14.815	3.446			l_2-loss	–	–
5	australian	logistic	**11.111**	**0.239**	12	a3a	logistic	**16.447**	**0.611**
		l_1-loss	13.333	42.818			l_1-loss	16.954	941.1448
		l_2-loss	12.222	265.842			l_2-loss	*	*
6	svmguide4	logistic	**0.000**	**0.005**	13	a4a	logistic	**16.448**	**0.672**
		l_1-loss	**0.000**	0.031			l_1-loss	17.090	1058.485
		l_2-loss	0.000	0.041			l_2-loss	*	*

Notes: – means out of memory. * means that the CPU time (t) is over 1 hour.

problem. Such nonlinear programming contains a set of nonlinear equality constraints and a simple lower bound constraint. To solve such NLP, we applied the smoothing Newton method proposed in [38] to solve the KKT system, which contains one pair of complementarity constraints. It is proven that the algorithm has a superlinear convergence rate. Extensive numerical results on the data sets from the LIBSVM library verified the efficiency of the proposed approach (SN) over almost all the data sets used in this paper, which can achieve competitive results while consuming less time than other methods, and strict local minimizers can be achieved both numerically and theoretically. The proposed approach has the potential to deal with other hyperparameter classification problems in SVM, which may involve multiple hyperparameters or multiple classes. These topics will be investigated further in the near future.

Notes

1. https://github.com/QingnaLi/GRCV-l1.
2. https://github.com/QingnaLi/GRCV-l2.
3. For data sets 11 and 14-20, GRCV-l_1 and GRCV-l_2 either take more than one hour to return a solution for some of them, or it is out of memory due to the large scale of the corresponding reformulated problem. Therefore, we did not report the numerical results here.

Acknowledgments

We would like to thank the editor for handling our submission. We would also like to thank the two anonymous referees for their valuable comments, which have helped us to further improve the paper.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This author's research is supported by the National Natural Science Foundation of China (NSFC) [grant number 12071032].

References

[1] Chauhan VK, Dahiya K, Sharma A. Problem formulations and solvers in linear SVM: a review. Artif Intell Rev. 2019;52(2):803–855. doi: 10.1007/s10462-018-9614-6
[2] Chapelle O, Vapnik V, Bousquet O, et al. Choosing multiple parameters for support vector machines. Mach Learn. 2002;46:131–159. doi: 10.1023/A:1012450327387
[3] Cortes C, Vapnik V. Support-vector networks. Mach Learn. 1995;20:273–297.
[4] Duan K, Keerthi SS, Poo AN. Evaluation of simple performance measures for tuning SVM hyperparameters. Neurocomputing. 2003;51:41–59. doi: 10.1016/S0925-2312(02)00601-X
[5] Keerthi S, Sindhwani V, Chapelle O. An efficient method for gradient-based adaptation of hyperparameters in SVM models. Adv Neural Inf Process Syst. 2007;19:673–680.
[6] Vapnik V. The nature of statistical learning theory. Berlin, Germany: Springer Science & Business Media; 2013.
[7] Dong YL, Xia ZQ, Wang MZ. An MPEC model for selecting optimal parameter in support vector machines. In: The First International Symposium on Optimization and Systems Biology; 2007. p. 351–357.
[8] Kunapuli G. A bilevel optimization approach to machine learning. NewYork: Rensselaer Polytechnic Institute; 2008.
[9] Kunapuli G, Bennett KP, Hu J, et al. Classification model selection via bilevel programming. Optim Methods Softw. 2008;23(4):475–489. doi: 10.1080/10556780802102586
[10] Kunapuli G, Bennett K, Hu J, et al. Bilevel model selection for support vector machines. In: CRM Proceedings and Lecture Notes; 2008, Vol. 45. p. 129–158.
[11] Momma M, Bennett KP. A pattern search method for model selection of support vector regression. In: Proceedings of the 2002 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics; 2002. p. 261–274.
[12] Bennett KP, Hu J, Ji X, et al. Model selection via bilevel optimization. In: The 2006 IEEE International Joint Conference on Neural Network Proceedings; 2006. p. 1922–1929.
[13] Li Q, Li Z, Zemkoho A. Bilevel hyperparameter optimization for support vector classification: theoretical analysis and a solution method. Mathematical Methods of Operations Research; 2022. p. 1–36.
[14] Kunisch K, Pock T. A bilevel optimization approach for parameter learning in variational models. SIAM J Imaging Sci. 2013;6(2):938–983. doi: 10.1137/120882706
[15] Colson B, Marcotte P, Savard G. An overview of bilevel optimization. Ann Oper Res. 2007;153:235–256. doi: 10.1007/s10479-007-0176-2
[16] Moore GM, Bergeron C, Bennett KP. Nonsmooth bilevel programming for hyperparameter selection. In: 2009 IEEE International Conference on Data Mining Workshops; 2009. p. 374–381.
[17] Okuno T, Takeda A, Kawana A, et al. On l_p-hyperparameter learning via bilevel nonsmooth optimization. J Mach Learn Res. 2021;22(1):11093–11139.

[18] Li Z, Qian Y, Li Q. A unified framework and a case study for hyperparameter selection in machine learning via bilevel optimization. In: 2022 5th International Conference on Data Science and Information Technology (DSIT); 2022. p. 1–8.
[19] Li Q, Qian Y, Zemkoho A. LP-Newton based global relaxation method for bilevel hyperparameter selection in support vector machines; 2023. doi: 10.48550/arXiv.2312.10848
[20] Coniglio S, Dunn A, Li Q, et al. Bilevel hyperparameter optimization for nonlinear support vector machines; 2023. p. 1–78.
[21] Nelder JA, Wedderburn RW. Generalized linear models. J R Stat Soc Ser A (General). 1972;135(3):370–384. doi: 10.2307/2344614
[22] Dayton CM. Logistic regression analysis. Stat. 1992;474:574.
[23] Franceschini F, Rafele C. Quality evaluation in logistic services. Int J Agile Manag Syst. 2000;2(1):49–54. doi: 10.1108/14654650010312589
[24] Glonek GF, McCullagh P. Multivariate logistic models. J R Stat Soc Ser B (Methodol). 1995;57(3):533–546. doi: 10.1111/j.2517-6161.1995.tb02046.x
[25] LaValley MP. Logistic regression. Circulation. 2008;117(18):2395–2399. doi: 10.1161/CIRCULATIONAHA.106.682658
[26] Abbasi E, Salehi F, Hassibi B. Sparse Covariance Estimation from Quadratic Measurements: A Precise Analysis. 2019 IEEE International Symposium on Information Theory (ISIT). 2019:2074–2078.
[27] Tripepi G, Jager KJ, Dekker FW, et al. Linear and logistic regression analysis. Kidney Int. 2008;73(7):806–810. doi: 10.1038/sj.ki.5002787
[28] Kleinbaum DG, Klein M, Pryor ER. Logistic regression: a self-learning text. 94, New York: springer; 2002.
[29] Nguyen T, Sanner S. Algorithms for direct 0–1 loss optimization in binary classification. In: International conference on machine learning. PMLR; 2013. p. 1085–1093.
[30] Abdalrada AS, Yahya OH, Alaidi AHM, et al. A predictive model for liver disease progression based on logistic regression algorithm. Period Eng Nat Sci. 2019;7(3):1255–1264.
[31] Stoltzfus JC. Logistic regression: a brief primer. Acad Emerg Med. 2011;18(10):1099–1104. doi: 10.1111/acem.2011.18.issue-10
[32] Meyer P. Bi-logistic growth. Technol Forecast Soc Change. 1994;47(1):89–102. doi: 10.1016/0040-1625(94)90042-6
[33] Walkling RA. Predicting tender offer success: a logistic analysis. J Finan Quant Analy. 1985;20(4):461–478. doi: 10.2307/2330762
[34] Pregibon D. Logistic regression diagnostics. Ann Stat. 1981;9(4):705–724. doi: 10.1214/aos/1176345513
[35] Peduzzi P, Concato J, Kemper E, et al. A simulation study of the number of events per variable in logistic regression analysis. J Clin Epidemiol. 1996;49(12):1373–1379. doi: 10.1016/S0895-4356(96)00236-3
[36] Chaudhuri K, Monteleoni C, Sarwate AD. Differentially private empirical risk minimization. J Mach Learn Res. 2011;12(3):1069–1109.
[37] Le T, Vo B, Fujita H, et al. A fast and accurate approach for bankruptcy forecasting using squared logistics loss with GPU-based extreme gradient boosting. Inf Sci. 2019;494:294–310. doi: 10.1016/j.ins.2019.04.060
[38] Liang L, Sun D, Toh KC. A squared smoothing Newton method for semidefinite programming; 2023. doi: 10.48550/arXiv.2303.05825
[39] Li T, Beirami A, Sanjabi M, et al. Tilted Empirical Risk Minimization. In: International Conference on Learning Representations; 2020. p. 1–44.
[40] Nocedal J, Wright SJ. Numerical optimization. New York, NY: Springer; 1999.

[41] Barrett R, Berry M, Chan TF, et al. Templates for the solution of linear systems: building blocks for iterative methods. Society for Industrial and Applied Mathematics; 1994. p. 1–18.
[42] Chen B, Harker PT. A non-interior-point continuation method for linear complementarity problems. SIAM J Matrix Anal Appl. 1993;14(4):1168–1190. doi: 10.1137/0614081
[43] Kanzow C. Some noninterior continuation methods for linear complementarity problems. SIAM J Matrix Anal Appl. 1996;17(4):851–868. doi: 10.1137/S0895479894273134
[44] Smale S. Algorithms for solving equations. In: The Collected Papers of Stephen Smale; 2000, Vol. 3. p. 1263–1286.
[45] Qi L, Sun D, Zhou G. A new look at smoothing newton methods for nonlinear complementarity problems and box constrained variational inequalities. Math Program. 2000;87:1–35. doi: 10.1007/s101079900127
[46] Sun D, Qi L. On NCP-functions. Comput Optim Appl. 1999;13:201–220. doi: 10.1023/A:1008669226453
[47] Pinar MC, Zenios SA. On smoothing exact penalty functions for convex constrained optimization. SIAM J Optim. 1994;4(3):486–511. doi: 10.1137/0804027
[48] Qi L, Sun J. A nonsmooth version of Newton's method. Math Program. 1993;58(1-3):353–367. doi: 10.1007/BF01581275
[49] Gill PE, Murray W, Saunders MA. SNOPT: an SQP algorithm for large-scale constrained optimization. SIAM Rev. 2005;47(1):99–131. doi: 10.1137/S0036144504446096
[50] Couellan N, Wang W. Bi-level stochastic gradient for large scale support vector machine. Neurocomputing. 2015;153:300–308. doi: 10.1016/j.neucom.2014.11.025
[51] Cokluk O. Logistic regression: concept and application. Educ Sci Theory Pract. 2010;10(3):1397–1407.