Mathematical Methods of Operations Research (2022) 96:315–350
https://doi.org/10.1007/s00186-022-00798-6

ORIGINAL ARTICLE

Bilevel hyperparameter optimization for support vector classification: theoretical analysis and a solution method

Qingna Li[1] · Zhen Li[2] · Alain Zemkoho[3]

Received: 13 October 2021 / Revised: 3 April 2022 / Accepted: 16 August 2022 /
Published online: 26 August 2022

Abstract

Support vector classification (SVC) is a classical and well-performed learning method for classification problems. A regularization parameter, which significantly affects the classification performance, has to be chosen and this is usually done by the cross-validation procedure. In this paper, we reformulate the hyperparameter selection problem for support vector classification as a bilevel optimization problem in which the upper-level problem minimizes the average number of misclassified data points over all the cross-validation folds, and the lower-level problems are the l_1-loss SVC problems, with each one for each fold in T-fold cross-validation. The resulting bilevel optimization model is then converted to a mathematical program with equilibrium constraints (MPEC). To solve this MPEC, we propose a global relaxation cross-validation algorithm (GR–CV) based on the well-know Sholtes-type global relaxation method (GRM). It is proven to converge to a C-stationary point. Moreover, we prove that the MPEC-tailored version of the Mangasarian–Fromovitz constraint qualification

Both authors contributed equally to this study

Q.Li: This author's research is supported by the National Science Foundation of China (NSFC) 12071032. A.Zemkoho: The work of this author is supported by the EPSRC grant EP/V049038/1 and the Alan Turing Institute under the EPSRC grant EP/N510129/1.

✉ Alain Zemkoho
a.b.zemkoho@soton.ac.uk

Qingna Li
qnl@bit.edu.cn

Zhen Li
lizhenbeili@126.com

1 School of Mathematics and Statistics/Beijing Key Laboratory on MCAACI/Key Laboratory of Mathematical Theory and Computation in Information Security, Beijing Institute of Technology, Beijing 100081, People's Republic of China

2 School of Mathematics and Statistics, Beijing Institute of Technology, Beijing 100081, People's Republic of China

3 School of Mathematical Sciences, University of Southampton, Southampton SO17 1BJ, UK

(MFCQ), which is a key property to guarantee the convergence of the GRM, automatically holds at each feasible point of this MPEC. Extensive numerical results verify the efficiency of the proposed approach. In particular, compared with other methods, our algorithm enjoys superior generalization performance over almost all the data sets used in this paper.

Keywords Support vector classification · Hyperparameter selection · Bilevel optimization · Mathematical program with equilibrium constraints · C-stationarity

Mathematics Subject Classification 90C33 · 90C90 · 49M20

1 Introduction

Support vector classification (SVC) is a classical and widely used learning method for classification problems; see, e.g., (Chauhan et al. 2019; Cortes and Vapnik 1995; Vapnik 2013). In SVC, the selection of hyperparameters, also known as hyperparameter selection, is a critical issue and has been addressed by many researchers both theoretically and practically (Chapelle et al. 2002; Dong et al. 2007; Duan et al. 2003; Keerthi et al. 2006; Kunapuli 2008; Kunapuli et al. 2008a, b). While there have been many interesting attempts to use bounds, gradient descent methods or other techniques to identify these hyperparameters (Chapelle et al. 2002; Duan et al. 2003; Keerthi et al. 2006), one of the most widely used methods is cross-validation (CV). A classical approach for cross-validation is the grid search method (Momma and Bennett 2002), where one needs to define a grid over the hyperparameters of interest, and search for the combination of hyperparameters that minimize the cross-validation error (CV error). Bennett et al. (2006) emphasize that one of the drawbacks of the grid search approach is that the continuity of the hyperparameter is ignored by the discretization. A formulation of the bilevel optimization model is proposed to choose hyperparameters (Bennett et al. 2006; Kunapuli 2008). Below, we will focus on the bilevel optimization approach which is the most relevant to our work. We refer to Yu and Zhu (2020), Luo (2016) for a survey of various hyperparameters optimization methods and applications.

In terms of selecting hyperparameters through bilevel optimization, different models and approaches have been considered in the literature. For example, Okuno et al. (2018) propose a bilevel optimization model to select the best hyperparameter for a nonsmooth, possibly nonconvex, l_p-regularized problem. They then present a smoothing-type algorithm with convergence analysis to solve this bilevel optimization model. Kunisch and Pock (2013) formulate a parameter learning problem for variational image denoising model into a bilevel optimization problem. They design a semismooth Newton's method for solving the resulting nonsmooth bilevel optimization problems. Moore et al. [17] develop an implicit gradient-type algorithm for selecting hyperparameters for linear SVM-type machine learning models which are expressed as bilevel optimization problems. Moore et al. (2009) propose a nonsmooth bilevel model to select hyperparameters for support vector regression (SVR) via T-fold cross-validation. They design a proximity control approximation algorithm to solve this bilevel optimization model. Couellan and Wang (2015) design a bilevel stochas-

tic gradient algorithm for training large scale SVM with automatic selection of the hyperparameter. We refer to Crockett and Fessler (2021), Colson et al. (2007), Dempe (2002), Dempe and Zemkoho (2020) for recent general surveys on bilevel optimization, as well as Mejía-de-Dios and Mezura-Montes (2019), Zemkoho and Zhou (2021), Fischer et al. (2021), Lin et al. (2014), Ye and Zhu (2010), Ochs et al. (2016, 2015) for some of the latest algorithms on the subject. Next, we provide a brief overview of the MPEC reformulation of the bilevel optimization problem, which will play a fundamental role in this paper.

For a bilevel program, replacing the lower-level problem by its Karush–Kuhn–Tucker (KKT) conditions will result in a mathematical program with equilibrium constraints (MPEC) Luo et al. (1996). Therefore, various algorithms for MPECs can be potentially applied to solve bilevel optimization problems, although one might want to pay attention to the fact that both problems are not necessarily equivalent. Bennett and her collaborators do a series of works (Bennett et al. 2006; Kunapuli et al. 2008b; Bennett et al. 2008; Kunapuli et al. 2008a; Kunapuli 2008) on hyperparameter selection by reformulating a bilevel program into an MPEC. For example, (Kunapuli et al. 2008b) considers a bilevel optimization model for selecting many hyperparameters for l_1-loss SVC problems, in which the upper-level problem has box constraints for the regularization parameter and feature selection. They reformulate this bilevel program into an MPEC and solve it by the inexact cross-validation method. Other methods include Newton-type algorithms (Wu et al. 2015; Harder et al. 2021; Lee et al. 2015).

Considering these works, a natural question is whether one can build up a bilevel hyperparameter selection for SVC? If yes, whether there are some special and hidden properties if we transfer the corresponding bilevel optimization problem to its corresponding MPEC and how we can solve it efficiently? This is the main motivation of the work in this paper.

In this paper, we consider a bilevel optimization model for selecting the hyperparameter in SVC. This regularization hyperparameter C is selected to minimize the T-fold cross-validated estimation of the out-of-sample misclassification error, which is basically a 0–1 loss function. Therefore, the upper-level problem minimizes the average misclassification error in T-fold cross-validation based on the optimal solution of the lower-level problem (we use the typical l_1-loss SVC model) for all the possible values of the hyperparameter C. There are several challenges to design efficient algorithms for such potentially large-scale bilevel programs. Firstly, the objective function in the upper-level problem is a 0–1 loss function, which is discontinuous and nonconvex. Secondly, the constraints for the upper-level problem involve the optimal solution set of the lower-level problem, i.e., the l_1-loss SVC model, for which the optimal solution is not explicitly given. To deal with the first challenge, we reformulate the minimization of the 0–1 loss function into a linear optimization problem inspired by the technique in Mangasarian (1994). We then replace the lower-level problem by its optimality conditions to tackle the second challenge. This therefore leads to an MPEC.

The contributions of the paper are as follows. Firstly, we propose a bilevel optimization model for hyperparameter selection in a binary SVC and study its reformulation as an MPEC. Secondly, we apply the GRM originating from Scholtes (2001) to solve this MPEC, which is shown to converge to a C-stationary point. The resulting algorithm

is called the GR–CV, which is a concrete implementation of the GRM for selecting the hyperparameter C in SVC. Thirdly, we prove the MPEC–Mangasarian–Fromovitz constraint qualification (MPEC–MFCQ, for short) property for each feasible point of our MPEC. The MPEC–MFCQ is a key property to guarantee the convergence of the GRM. We show that it automatically holds for our problem thanks to its special structure. Finally, we conduct extensive numerical experiments, which show that our method is very efficient; in particular, it enjoys superior generalization performance over almost all the data sets used in this paper.

The paper is organized as follows. In Sect. 2, based on T-fold cross-validation for SVC, we introduce a bilevel optimization model to select an optimal hyperparameter for SVC. We also analyze the interesting properties of the lower-level problem. In Sect. 3, we reformulate the bilevel optimization problem as an MPEC (also known as the KKT reformulation), and apply the GRM for solving the MPEC. In Sect. 4, we prove that every feasible point of this MPEC satisfies the regularity condition MPEC–MFCQ, which is a key property to guarantee the convergence of the GRM. In Sect. 5, we present some computational experiments comparing the resulting GR–CV based on the GRM with two other ones, which have been used in the literature for a similar purpose; i.e., the inexact cross-validation method (In–CV) and the grid search method (G–S). We conclude the paper in Sect. 6.

Notations. For $x \in \mathbb{R}^n$, $\|x\|_0$ denotes the number of nonzero elements in x, while $\|x\|_1$ and $\|x\|_2$ correspond to the l_1-norm and l_2-norm of x, respectively. Also, we will use $x_+ = ((x_1)_+, \cdots, (x_n)_+) \in \mathbb{R}^n$, where $(x_i)_+ = \max(x_i, 0)$. $|\Omega|$ denotes the number of elements in the set $\Omega \subset \mathbb{R}^n$. We use $\mathbf{1}_k$ to denote a vector with elements all ones in $\mathbb{R}^k$. I_k is the identity matrix in $\mathbb{R}^{k\times k}$, while e_γ^k is the γ-th row vector of an identity matrix in $\mathbb{R}^{k\times k}$. The notation $\mathbf{0}_{k\times q}$ represents a zero matrix in $\mathbb{R}^{k\times q}$ and $\mathbf{0}_k$ stands for a zero vector in $\mathbb{R}^k$. On the other hand, $\mathbf{0}_{(\tau,\,\kappa)}$ will be used for a submatrix of the zero matrix, where τ is the index set of the rows and κ is the index set of the columns. Similarly to the case of zero matrix, $I_{(\tau,\,\tau)}$ corresponds to a submatrix of an identity matrix indexed by both rows and columns in the set τ. Finally, $\Theta_{(\tau,\,\cdot)}$ represents a submatrix of the matrix Θ, where τ is the index set of the rows, and x_τ is a subvector of the vector x corresponding to the index set τ.

2 Bilevel hyperparameter optimization for SVC

We start this section by first introducing the problem settings in relation to the T-fold cross-validation for SVC. Subsequently, we present the lower-level problem with some interesting and relevant properties for further analysis in the later parts of the paper. Finally, we introduce the upper-level problem, that is, the bilevel optimization model for hyperparameter selection in SVC.

2.1 T-fold cross-validation for SVC

As discussed in the introduction, the most commonly used method for selecting the hyperparameter C is T-fold cross-validation. In T-fold cross-validation, the data set is

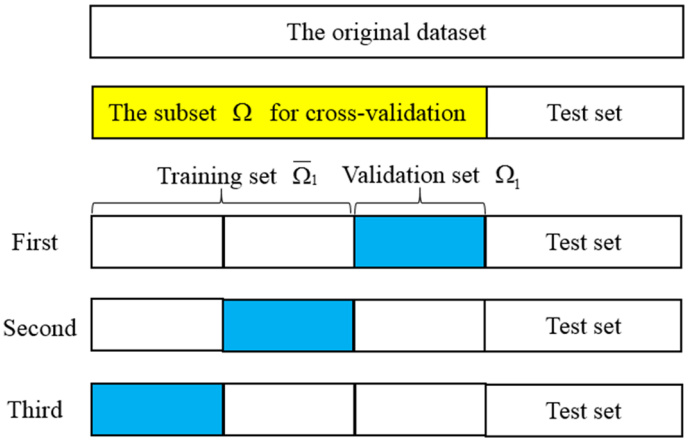


Fig. 1 Three-fold cross-validation

split into a subset Ω with l_1 points, which is used for cross-validation, and a hold-out test set Θ with l_2 points. Here, $\Omega = \{(x_i, y_i)\}_{i=1}^{l_1} \in \mathbb{R}^{n+1}$, where $x_i \in \mathbb{R}^n$ denotes a data point and $y_i \in \{\pm 1\}$ the corresponding label. For T-fold cross-validation, Ω is equally partitioned into T disjoint subsets[1], as done in Couellan and Wang (2015), Moore et al. (2009), one for each fold. The process is executed T iterations. For the t-th iteration ($t = 1, \ldots, T$), the t-th fold is the validation set Ω_t, and the remaining $T - 1$ folds make up the training set $\overline{\Omega}_t = \Omega \backslash \Omega_t$. Therefore, in the t-th iteration, the separating hyperplane is trained using the training set $\overline{\Omega}_t$, and the validation error is computed on the validation set Ω_t.

Then, the cross-validation error (CV error) is the average of the validation error over all the T iterations. The value of C that gives the best CV error will be selected. Finally, the final classifier is trained using all the data in Ω and the rescaled optimal C. The test error is computed on the test set Θ. Note that the CV error and the test error are the evaluation indices for the classification performance in T-fold cross-validation. As shown in Fig. 1, for three-fold cross-validation, the yellow part is the subset Ω which is used for three-fold cross-validation. In the first iteration, the blue part is the validation set Ω_1, and the remaining two folds are the training set $\overline{\Omega}_1$. The second and third iterations have similar meanings.

Let m_1 be the size of the validation set Ω_t and m_2 the size of the training set $\overline{\Omega}_t$. The corresponding index sets for the validation and training sets are $\mathcal{N}_t$ and $\overline{\mathcal{N}}_t$, respectively. In T-fold cross-validation, there are T validation sets. Therefore, there are totally Tm_1 validation points in T-fold cross-validation. We use the index set

$$Q_u := \{i \mid i = 1, 2, \cdots, Tm_1\} \tag{1}$$

[1] Actually, Ω can be partitioned unequally, as done in Bennett et al. (2006), Lee et al. (2015), for example. Our analysis here applies to unequal partitions of Ω as well. However, to demonstrate it easily, we use an equal partition of Ω.

to represent all the validation points in T-fold cross-validation. Similarly, there are totally Tm_2 training points in T-fold cross-validation. We use the index set

$$Q_l := \{i \mid i = 1, 2, \cdots, Tm_2\} \tag{2}$$

to represent all the training points in T-fold cross-validation. These two index sets will be used later.

To analyze different cases of the data points in the training set and the validation set, we need to introduce the soft-margin support vector classification (without bias term) Cristianini and Shawe-Taylor (2000), Galli and Lin (2021). The traditional SVC model which is referred to as hard-margin SVC requires that the data should be strictly separated, i.e., the constraints must be satisfied strictly. However, the regularized model (soft-margin SVC) allows that the data could be wrongly labelled, i.e., the inequality constraints can be violated, which is the case in the model

$$\min_{w\in\mathbb{R}^n} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{\tau} \xi(w; \quad x_i, y_i), \tag{3}$$

where $C \geq 0$ is a penalty parameter and $\xi(\cdot)$ is the loss function. If $\xi(w; \quad x_i, y_i) = (1 - y_i(x_i^\top w))_+$, it is referred to as the l_1-loss function; if $\xi(w; \quad x_i, y_i) = (1 - y_i(x_i^\top w))_+^2$, it is referred to as the l_2-loss function. We refer to Chauhan et al. (2019), Wang et al. (2021), Huang et al. (2013) for various other types of loss functions. Next, we use Fig. 2 to show geometric relationships of different cases in soft-margin SVC.

For a sample (x_i, y_i), the point x_i is referred to as a positive point if $y_i = 1$; the point x_i is referred to as a negative point if $y_i = -1$. In Fig. 2, the plus signs '+' are the positive points (i.e., $y_i = 1$) and the minus signs '−' are the negative ones (i.e., $y_i = -1$). The distance between the hyperplanes $H_1 : w^\top x = 1$ and $H_2 : w^\top x = -1$ is called *margin*. The *separating hyperplane* H lies between H_1 and H_2. Clearly, the hyperplanes H_1 and H_2 are the boundaries of the margin. Therefore, if a positive point lies on the hyperplane H_1 or a negative point lies on the hyperplane H_2, we call it lying on the boundary of the margin (indicated by '①' in Fig. 2). If a positive point lies between the separating hyperplane H and the hyperplane H_1, or a negative point lies between the separating hyperplane H and the hyperplane H_2, we call it lying between the separating hyperplane H and the boundary of the margin (indicated by '②' in Fig. 2). Similarly, if a positive point lies on the correctly classified side of the hyperplane H_1, or a negative point lies on the correctly classified side of the hyperplane H_2, we call it lying on the correctly classified side of the boundary of the margin (indicated by '③' in Fig. 2).

Based on Fig. 2, we have the following observations which address different cases for the data points in the training set $\overline{\Omega}_t$. Consider the soft-margin SVC problem corresponding to the t-th fold, i.e., the t-th training set $\overline{\Omega}_t$ and validation set Ω_t are used. We also use w^t to represent the optimal solution in (3) trained by $\overline{\Omega}_t$.

Proposition 1 *Let w^t be an optimal solution of the t-th soft-margin SVC model. For $i \in \overline{\mathcal{N}}_t$, consider a positive point x_i. Then it holds that:*

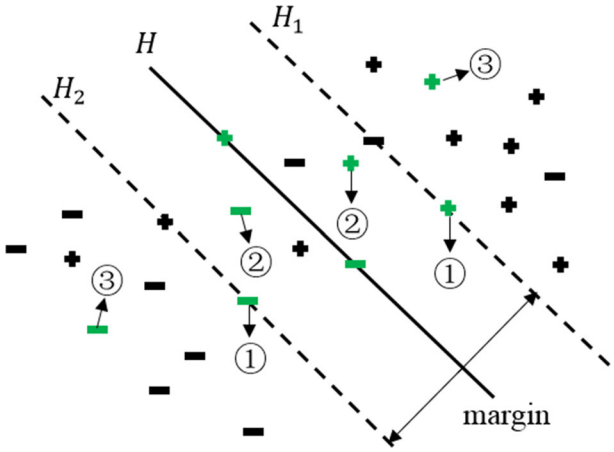


Fig. 2 Training points in soft-margin support vector machine

(a) x_i *satisfies* $(w^t)^\top x_i < 0$ *if and only if it lies on the misclassified side of the separating hyperplane* H*, and is therefore misclassified.*
(b) x_i *satisfies* $(w^t)^\top x_i = 0$ *if and only if it lies on the separating hyperplane* H*, and is therefore correctly classified.*
(c) x_i *satisfies* $0 < (w^t)^\top x_i < 1$ *if and only if it lies between the separating hyperplane* H *and the boundary of the margin; hence, it is correctly classified.*
(d) x_i *satisfies* $(w^t)^\top x_i = 1$ *if and only if it lies on the boundary of the margin, and is therefore correctly classified.*
(e) x_i *satisfies* $(w^t)^\top x_i > 1$ *if and only if it lies on the correctly classified side of the boundary of the margin, and is therefore correctly classified.*

A result analogous to Proposition 1 can be stated for the negative points. In Fig. 3, any point $x_i \in \overline{\Omega}_t$ in blue is a training point in each case (notation is the same as in Fig. 5).

As for data points in the validation set Ω_t, we have the following scenarios.

Proposition 2 *Let* w^t *be an optimal solution of the* t*-th soft-margin SVC model. For* $i \in \mathcal{N}_t$*, consider a positive point* x_i*. Then it holds that:*

(a) x_i *satisfies* $(w^t)^\top x_i < 0$ *if and only if it lies on the misclassified side of the separating hyperplane* H*, and is therefore misclassified.*
(b) x_i *satisfies* $(w^t)^\top x_i = 0$ *if and only if it lies on the separating hyperplane* H*, and is therefore correctly classified.*
(c) x_i *satisfies* $(w^t)^\top x_i > 0$ *if and only if it lies on the correctly classified side of the separating hyperplane* H*, and it is hence correctly classified.*

A result analogous to Proposition 2 can be stated for the negative points. In Fig. 4, any point $x_i \in \Omega_t$ in blue is a validation point in each case (notation is the same as in Fig. 6).

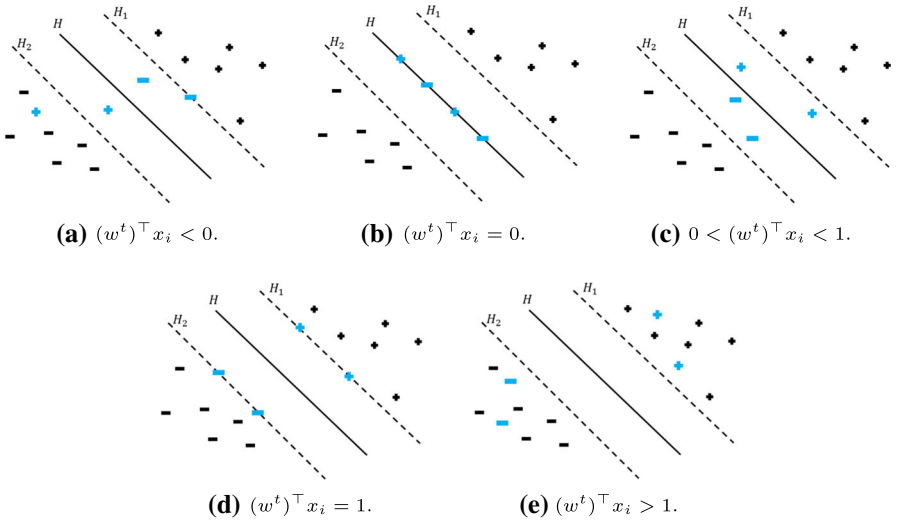


Fig. 3 Each case for different values of $(w^t)^\top x_i$ with x_i in the training set $\overline{\Omega}_t$

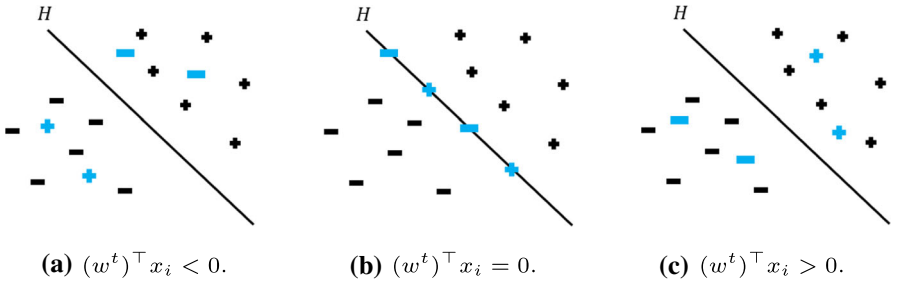


Fig. 4 Each case for different values of $(w^t)^\top x_i$ with x_i in the validation set Ω_t

Remark 1 Note that Propositions 1 and 2 are applicable to the soft-margin SVC model with other loss functions. These two propositions will be used in the proof of Propositions 3 and 4.

2.2 The lower-level problem

In this part, we focus on the lower-level problem. That is, given hyperparameter C and the training set $\overline{\Omega}_t$, we train the dataset via l_1-loss SVC model. We will also discuss the properties of the lower-level problem.

2.2.1 The training model: l_1-loss SVC

In T-fold cross-validation, there are T lower-level problems. In the t-th lower-level problem, we train the t-th fold training set $\overline{\Omega}_t$ by the soft-margin SVC model in (3)

with the l_1-loss function[2]. That is, given $C \geq 0$, we solve the following optimization problem:

$$\min_{w^t \in \mathbb{R}^n} \frac{1}{2}\|w^t\|_2^2 + C \sum_{i \in \overline{\mathcal{N}}_t} (1 - y_i(x_i^\top w^t))_+.$$

A popular reformulation of the problem above is the convex quadratic optimization problem obtained by introducing slack variables $\xi^t \in \mathbb{R}^{m_2}$:

$$\begin{array}{ll} \min\limits_{w^t \in \mathbf{R}^n,\, \xi^t \in \mathbf{R}^{m_2}} & \frac{1}{2}\left\|w^t\right\|_2^2 + C \sum\limits_{i=1}^{m_2} \xi_i^t \\ \text{s.t.} & B^t w^t \geq \mathbf{1} - \xi^t, \\ & \xi^t \geq \mathbf{0}, \end{array} \tag{4}$$

where, for $t = 1, \cdots, T$ and $k = m_1 + 1, \cdots, l_1$, we have

$$B^t = \begin{bmatrix} y_{t_{m_1+1}} x_{t_{m_1+1}}^\top \\ \vdots \\ y_{t_{l_1}} x_{t_{l_1}}^\top \end{bmatrix} \in \mathbb{R}^{m_2 \times n}, \quad (x_{t_k}, y_{t_k}) \in \overline{\Omega}_t,$$

and we use ξ_i^t to denote the i-th element of $\xi^t \in \mathbb{R}^{m_2}$.

Let $\alpha^t \in \mathbb{R}^{m_2}$ and $\mu^t \in \mathbb{R}^{m_2}$ be the multipliers of the constraints in (4). We can write the KKT conditions for the lower-level problem (4) as

$$\mathbf{0} \leq \alpha^t \perp B^t w^t - \mathbf{1} + \xi^t \geq \mathbf{0}, \tag{5a}$$

$$\mathbf{0} \leq \xi^t \perp \mu^t \geq \mathbf{0}, \tag{5b}$$

$$w^t - (B^t)^\top \alpha^t = \mathbf{0}, \tag{5c}$$

$$C\mathbf{1} - \alpha^t - \mu^t = \mathbf{0}, \tag{5d}$$

where for two vectors a and b, writing $\mathbf{0} \leq a \perp b \geq \mathbf{0}$ means that we have $a^\top b = 0$, $a \geq \mathbf{0}$ and $b \geq \mathbf{0}$. Also note that each complementary constraint in (5a) corresponds to a training point x_i with $i \in Q_l$ in (2). Each training point corresponds to a slack variable ξ_i^t. So each complementary constraint in (5b) corresponds to a training point x_i with $i \in Q_l$ in (2). Therefore, there is a one-to-one correspondence between the index set of the training points Q_l and the complementary constraints in (5a) and (5b), respectively. This will be used in the definition of some index sets below.

Furthermore, we would like to emphasize the support vectors implied in (5). From (5c), the weight vector $w^t = (B^t)^\top \alpha^t = \sum\limits_{i \in \overline{\mathcal{N}}_t} \alpha_i^t y_i x_i$. It implies that only the data

[2] We choose the l_1-loss SVC model as the typical lower-level problem due to the following reasons. Firstly, from the practical perspective, the l_1-loss SVC model is a widely used statistical model in machine learning (Yan and Li 2020; Zhang 2004; Shalev-Shwartz et al. 2011). Secondly, the l_1-loss SVC model is more challenging to tackle than the l_2-loss SVC model due to the nonsmoothness of the l_1-loss function

points $x_i \in \overline{\Omega}_t$ which correspond to $\alpha_i^t \neq 0$ are involved. By $\alpha_i^t \geq 0$ in (5a), it means that only $x_i \in \overline{\Omega}_t$ with $\alpha_i^t > 0$ are involved. It is for this reason that they are called *support vectors*. By eliminating μ^t and w^t from the system in (5), we get the reduced KKT conditions for problem (4) as follows:

$$\begin{cases} \mathbf{0} \leq \alpha^t \perp B^t (B^t)^\top \alpha^t - \mathbf{1} + \xi^t \geq \mathbf{0}, \\ \mathbf{0} \leq \xi^t \perp C\mathbf{1} - \alpha^t \geq \mathbf{0}. \end{cases} \tag{6}$$

2.2.2 Some properties of the lower-level problem

Let $\alpha \in \mathbb{R}^{Tm_2}$, $\xi \in \mathbb{R}^{Tm_2}$, $w \in \mathbb{R}^{Tn}$, and $B \in \mathbb{R}^{Tm_2 \times Tn}$ be defined by

$$\alpha := \begin{bmatrix} \alpha^1 \\ \alpha^2 \\ \vdots \\ \alpha^T \end{bmatrix}, \ \xi := \begin{bmatrix} \xi^1 \\ \xi^2 \\ \vdots \\ \xi^T \end{bmatrix}, \ w := \begin{bmatrix} w^1 \\ w^2 \\ \vdots \\ w^T \end{bmatrix}, \text{ and } B := \begin{bmatrix} B^1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & B^2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & B^T \end{bmatrix}, \tag{7}$$

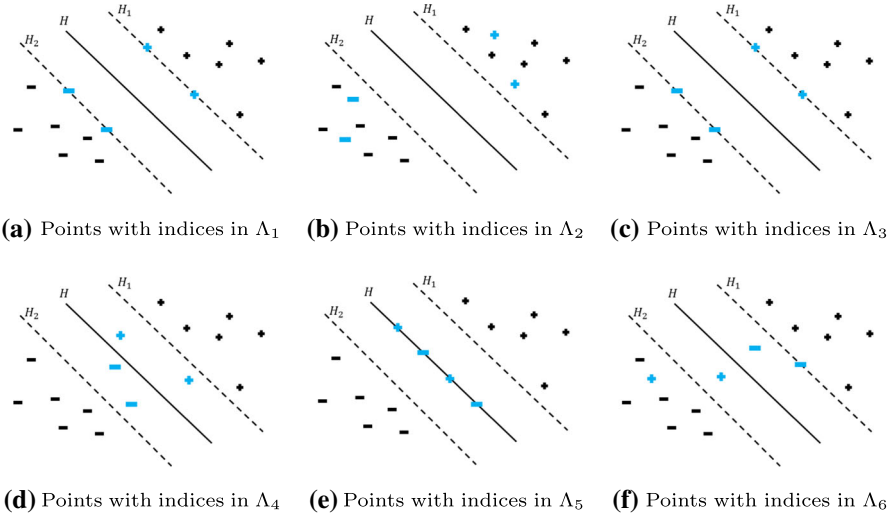
respectively. The KKT conditions in (6) can be decomposed as

$$\Lambda_1 := \{i \in Q_l \mid \alpha_i = 0, \ (BB^\top \alpha - \mathbf{1} + \xi)_i = 0, \ \xi_i = 0\}, \tag{8}$$
$$\Lambda_2 := \{i \in Q_l \mid \alpha_i = 0, \ (BB^\top \alpha - \mathbf{1} + \xi)_i > 0, \ \xi_i = 0\}, \tag{9}$$
$$\Lambda_3 := \{i \in Q_l \mid 0 < \alpha_i \leq C, \ (BB^\top \alpha - \mathbf{1} + \xi)_i = 0, \ \xi_i = 0\}, \tag{10}$$
$$\Lambda_4 := \{i \in Q_l \mid \alpha_i = C, \ (BB^\top \alpha - \mathbf{1} + \xi)_i = 0, \ 0 < \xi_i < 1\}, \tag{11}$$
$$\Lambda_5 := \{i \in Q_l \mid \alpha_i = C, \ (BB^\top \alpha - \mathbf{1} + \xi)_i = 0, \ \xi_i = 1\}, \tag{12}$$
$$\Lambda_6 := \{i \in Q_l \mid \alpha_i = C, \ (BB^\top \alpha - \mathbf{1} + \xi)_i = 0, \ \xi_i > 1\}. \tag{13}$$

Obviously, the intersection of any pair of these index sets Λ_i for $i = 1, \ldots, 6$ is empty. An illustrative representation of data points corresponding to these index sets is given in Fig. 5.

Proposition 3 *Considering the training points corresponding to Q_l in (2), let (α, ξ) satisfy the conditions in (6). Then, the following statements hold true:*

(a) *The points $\{x_i\}_{i \in \Lambda_1}$ lie on the boundary of the margin; they are correctly classified points, but are not support vectors.*
(b) *The points $\{x_i\}_{i \in \Lambda_2}$ lie on the correctly classified side of the boundary of the margin; they are correctly classified points, but are not support vectors.*
(c) *The points $\{x_i\}_{i \in \Lambda_3}$ lie on the boundary of the margin; they are correctly classified points and are support vectors.*
(d) *The points $\{x_i\}_{i \in \Lambda_4}$ lie between the separating hyperplane H and the boundary of the margin; they are correctly classified therefore support vectors.*
(e) *The points $\{x_i\}_{i \in \Lambda_5}$ lie on the separating hyperplane H; they are correctly classified points and are support vectors.*
(f) *The points $\{x_i\}_{i \in \Lambda_6}$ lie on the misclassified side of the separating hyperplane H; they are misclassified points and are support vectors.*



(a) Points with indices in Λ_1 **(b)** Points with indices in Λ_2 **(c)** Points with indices in Λ_3

(d) Points with indices in Λ_4 **(e)** Points with indices in Λ_5 **(f)** Points with indices in Λ_6

Fig. 5 Representation of points with index sets Λ_j, $j = 1, \ldots, 6$

Proof We take positive points for example. The same analysis can be applied to negative ones. Since $w = B^\top \alpha$ in (5c), we get $(BB^\top \alpha - \mathbf{1} + \xi)_i = (Bw - \mathbf{1} + \xi)_i$.

(a) For the points $\{x_i\}_{i\in\Lambda_1}$, since $\xi_i = 0$ in (8), we have $(Bw-\mathbf{1}+\xi)_i = (Bw-\mathbf{1})_i = 0$, that is, $y_i(w^\top x_i) - 1 = 0$. For a positive point, $y_i = 1$, it implies that $w^\top x_i = 1$. It corresponds to (d) in Proposition 1. Therefore, it means that the point x_i lies on the boundary of the margin. It is correctly classified, and it is not a support vector, since $\alpha_i = 0$.

(b) For the points $\{x_i\}_{i\in\Lambda_2}$, since $\xi_i = 0$ in (9), we have $(Bw-\mathbf{1}+\xi)_i = (Bw-\mathbf{1})_i > 0$, that is, $y_i(w^\top x_i) - 1 > 0$. For a positive point, $y_i = 1$, it implies that $w^\top x_i > 1$. It corresponds to (e) in Proposition 1. Therefore, it means that the point x_i lies on the correctly classified side of the boundary of the margin. It is correctly classified, but not a support vector, as $\alpha_i = 0$.

(c) For the points $\{x_i\}_{i\in\Lambda_3}$, since $\xi_i = 0$ in (10), we have $(Bw-\mathbf{1}+\xi)_i = (Bw-\mathbf{1})_i = 0$, that is, $y_i(w^\top x_i) - 1 = 0$. For a positive point, $y_i = 1$, it implies that $w^\top x_i = 1$. It corresponds to (d) in Proposition 1. Therefore, it means that the point x_i lies on the boundary of the margin. It is correctly classified, and it is a support vector, since $\alpha_i > 0$.

(d) For the points $\{x_i\}_{i\in\Lambda_4}$, since $0 < \xi_i < 1$ in (11), we have $0 < (Bw)_i < 1$, that is, $0 < y_i(w^\top x_i) < 1$. For a positive point, $y_i = 1$, it implies that $0 < w^\top x_i < 1$. It corresponds to (c) in Proposition 1. Therefore, x_i lies between the separating hyperplane H and the boundary of the margin. It is correctly classified, and it is a support vector, since $\alpha_i > 0$.

(e) For the points $\{x_i\}_{i\in\Lambda_5}$, since $\xi_i = 1$ in (12), we have $(Bw-\mathbf{1}+\xi)_i = (Bw)_i = 0$, that is, $y_i(w^\top x_i) = 0$. For a positive point, $y_i = 1$, it implies that $w^\top x_i = 0$. It corresponds to (b) in Proposition 1. Therefore, it means that the point x_i lies on

the separating hyperplane H. It is correctly classified, and it is a support vector, since $\alpha_i > 0$.

(f) For the points $\{x_i\}_{i\in\Lambda_6}$, since $\xi_i > 1$ in (13), we have $(Bw)_i < 0$, that is, $y_i(w^\top x_i) < 0$. For a positive point, $y_i = 1$, it implies that $w^\top x_i < 0$. It corresponds to (a) in Proposition 1. Therefore, it means that the point x_i lies on the misclassified side of the separating hyperplane H. It is misclassified, and it is a support vector, since $\alpha_i > 0$.

Remark 2 Note that all the data points x_i for $i \in \Lambda_1$ corresponding to Fig. 5a and $i \in \Lambda_3$ corresponding to Fig. 5c lie on the boundary of the margin. In other words, Fig. 5a and Fig. 5c are identical. However, the values of α_i for $i \in \Lambda_1$ and $i \in \Lambda_3$ are different, so we demonstrate them in two subfigures.

2.3 The upper-level problem

In this part, we introduce the upper-level problem, that is, the bilevel optimization model for hyperparameter selection in SVC under the settings of T-fold cross-validation. Note that the aim of the upper-level problem is to minimize the T-fold CV error measured on the validation sets based on the optimal solutions of the lower-level problems. Specifically, the basic bilevel optimization model for selecting the hyperparameter C in SVC is formulated as

$$\begin{aligned} \min_{C\in\mathbb{R},\ w^t\in\mathbb{R}^n,\ t=1,\cdots,T} \ & \frac{1}{T}\sum_{t=1}^{T}\frac{1}{m_1}\sum_{i\in\mathcal{N}_t} \| \left(-y_i\left(x_i^\top w^t\right)\right)_+ \|_0 \\ \text{s.t.} \quad & C \geq 0, \\ & \text{and for} \quad t = 1,\cdots,T: \\ & w^t \in \operatorname*{argmin}_{w\in\mathbb{R}^n} \left\{ \frac{1}{2}\|w\|_2^2 + C\sum_{i\in\overline{\mathcal{N}}_t}\left(1 - y_i\left(x_i^\top w\right)\right)_+ \right\}. \end{aligned} \tag{14}$$

Here, the expression $\sum_{i\in\mathcal{N}_t} \| \left(-y_i\left(x_i^\top w^t\right)\right)_+ \|_0$ basically counts the number of data points that are misclassified in the validation set Ω_t, while the outer summation (i.e., the objective function in (14)) averages the misclassification error over all the folds.

Problem (14) can be equivalently written in the matrix form as follows

$$\begin{aligned} \min_{C\in\mathbb{R},\ w^t\in\mathbb{R}^n,\ t=1,\cdots,T} \ & \frac{1}{T}\sum_{t=1}^{T}\frac{1}{m_1} \| \left(-A^t w^t\right)_+ \|_0 \\ \text{s.t.} \quad & C \geq 0, \\ & \text{and for} \quad t = 1,\cdots,T: \\ & w^t \in \operatorname*{argmin}_{w\in\mathbb{R}^n} \left\{ \frac{1}{2}\|w\|_2^2 + C\| \left(\mathbf{1} - B^t w\right)_+ \|_1 \right\}, \end{aligned} \tag{15}$$

where, for $t = 1, \cdots, T$ and $k = 1, \cdots, m_1$, we have

$$A^t = \begin{bmatrix} y_{t_1} x_{t_1}^\top \\ \vdots \\ y_{t_{m_1}} x_{t_{m_1}}^\top \end{bmatrix} \in \mathbb{R}^{m_1 \times n} \quad \text{and} \quad (x_{t_k}, y_{t_k}) \in \Omega_t.$$

Remark 3 Compared with the model in Kunapuli et al. (2008b), we consider a simpler bilevel optimization model, only with an extra constraint $C \geq 0$ in the upper-level problem.

3 Single-level reformulation and method

In this section, we first reformulate the bilevel optimization problem as a single-level optimization problem, precisely, we write the problem as an MPEC. Then we present the properties of this single-level problem. Finally, we discuss the GRM to solve the MPEC problem.

3.1 The MPEC reformulation

Recall the upper-level objective function in (15) is a measure of misclassification error based on the T out-of-sample validation sets, which we minimize. The measure used here is the classical CV error for classification, the average number of the data points misclassified. It is clear that $\|(\cdot)_+\|_0$ is discontinuous and nonconvex. However, the function $\|(\cdot)_+\|_0$ can be characterized as the minimum of the sum of all elements of the solution to the following linear optimization problem as demonstrated in Mangasarian (1994), i.e.,

$$\|r_+\|_0 = \left\{ \min \sum_{i=1}^{m_1} \zeta_i \,:\, \zeta = \underset{u}{\operatorname{argmin}} \left\{ -u^\top r \;:\; \mathbf{0} \leq u \leq \mathbf{1} \right\} \right\}.$$

Therefore, for each fold, $\| \left(-A^t w^t\right)_+ \|_0$ is the minimum of the sum of all elements of the solution to the following linear optimization problem:

$$\begin{aligned} \min_{\zeta^t \in \mathbb{R}^{m_1}} \quad & -(\zeta^t)^\top (-A^t w^t) \\ \text{s.t.} \quad & \zeta^t \geq \mathbf{0}, \\ & \mathbf{1} - \zeta^t \geq \mathbf{0}. \end{aligned} \tag{16}$$

Let $\hat{\zeta}^t$ be the solution of problem (16) such that $\sum_{i=1}^{m_1} \hat{\zeta}_i^t$ is the minimum of the sum of all elements of the solution to problem (5). This implies that $\| \left(-A^t w^t\right)_+ \|_0 = \sum_{i=1}^{m_1} \zeta_i^t$ in each fold. According to Proposition 2, there are two cases for the validation points:

1. If the validation point $(x_i, y_i) \in \Omega_t$ is misclassified, then $y_i \left(x_i^\top w^t\right) < 0$. That is, $(-A^t w^t)_i > 0$, which corresponds to $(\left(-A^t w^t\right)_+)_i > 0$.
2. If the validation point $(x_i, y_i) \in \Omega_t$ is correctly classified, we have $y_i \left(x_i^\top w^t\right) \geq 0$. There are two cases. Firstly, x_i lies on the separating hyperplane H, that is, $y_i \left(x_i^\top w^t\right) = 0$. For $y_i = 1$, there is $(-A^t w^t)_i = 0$, which corresponds to $(\left(-A^t w^t\right)_+)_i = 0$. Secondly, x_i lies on the correctly classified side of the separating hyperplane H, that is, $y_i \left(x_i^\top w^t\right) > 0$. For $y_i = 1$, there is $(-A^t w^t)_i < 0$, which corresponds to $(\left(-A^t w^t\right)_+)_i = 0$.

Combining with $\| \left(-A^t w^t\right)_+ \|_0 = \sum_{i=1}^{m_1} \hat{\zeta}_i^t$, it means that

$$\hat{\zeta}_i^t = \begin{cases} 1, & \text{if } (x_i, y_i) \in \Omega_t \text{ is misclassified,} \\ 0, & \text{if } (x_i, y_i) \in \Omega_t \text{ is correctly classified,} \end{cases} \tag{17}$$

where $\hat{\zeta}_i^t$ is the i-th element of $\hat{\zeta}^t$ in the t-th fold.

The linear programs (LPs) (16), for $t = 1, \cdots, T$, are inserted into the bilevel optimization problem in order to recast the discontinuous upper-level objective function into a continuous one. Each LP in the form of (16) can also be replaced with its KKT conditions as follows

$$\begin{cases} \mathbf{0} \leq \zeta^t \perp \lambda^t \geq \mathbf{0}, \\ \mathbf{0} \leq z^t \perp \mathbf{1} - \zeta^t \geq \mathbf{0}, \\ A^t w^t - \lambda^t + z^t = \mathbf{0}. \end{cases}$$

By eliminating λ^t and w^t with $w^t = (B^t)^\top \alpha^t$ in (5c), we get the reduced KKT conditions for problem (16) with

$$\mathbf{0} \leq \zeta^t \perp A^t (B^t)^\top \alpha^t + z^t \geq \mathbf{0}, \tag{18a}$$

$$\mathbf{0} \leq z^t \perp \mathbf{1} - \zeta^t \geq \mathbf{0}. \tag{18b}$$

Note that each complementary constraint in (18a) corresponds to a validation point x_i with $i \in Q_u$ in (1). Each validation point corresponds to a variable ζ_i^t. So we have each complementary constraint in (18b) corresponds to a validation point x_i with $i \in Q_u$ in (1). Therefore, there is a one-to-one correspondence between the index set of the validation points Q_u and the complementary constraints in (18a) and (18b), respectively.

Combining the systems in (6) and (18), we can transform the bilevel optimization problem (15) into the single-level optimization problem

$$\begin{aligned} \min_{\substack{C\in\mathbb{R}\\ \zeta^t\in\mathbb{R}^{m_1},\ z^t\in\mathbb{R}^{m_1}\\ \alpha^t\in\mathbb{R}^{m_2},\ \xi^t\in\mathbb{R}^{m_2}\\ t=1,\ \cdots,\ T}} \quad & \frac{1}{Tm_1}\sum_{i=1}^{m_1}\sum_{t=1}^{T}\zeta_i^t \\ \text{s.t.} \quad & C\geq 0, \\ & \text{and for} \quad t=1,\ \cdots,\ T: \\ & \begin{cases} \mathbf{0}\leq \zeta^t \perp A^t(B^t)^\top\alpha^t + z^t \geq \mathbf{0}, \\ \mathbf{0}\leq z^t \perp \mathbf{1}-\zeta^t \geq \mathbf{0}, \\ \mathbf{0}\leq \alpha^t \perp B^t(B^t)^\top\alpha^t - \mathbf{1}+\xi^t \geq \mathbf{0}, \\ \mathbf{0}\leq \xi^t \perp C\mathbf{1}-\alpha^t \geq \mathbf{0}. \end{cases} \end{aligned} \tag{19}$$

Note that the constraints $C\mathbf{1}-\alpha^t\geq\mathbf{0}$ and $\alpha^t\geq\mathbf{0}$ imply $C\geq 0$. Therefore, we remove the redundant constraint $C\geq 0$, and get an equivalent form of the problem above as follows

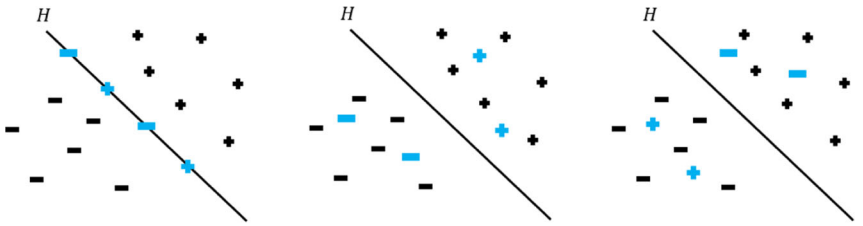
$$\begin{aligned} \min_{\substack{C\in\mathbb{R}\\ \zeta^t\in\mathbb{R}^{m_1},\ z^t\in\mathbb{R}^{m_1}\\ \alpha^t\in\mathbb{R}^{m_2},\ \xi^t\in\mathbb{R}^{m_2}\\ t=1,\ \cdots,\ T}} \quad & \frac{1}{Tm_1}\sum_{i=1}^{m_1}\sum_{t=1}^{T}\zeta_i^t \\ \text{s.t.} \quad & \text{for} \quad t=1,\ \cdots,\ T: \\ & \begin{cases} \mathbf{0}\leq \zeta^t \perp A^t(B^t)^\top\alpha^t + z^t \geq \mathbf{0}, \\ \mathbf{0}\leq z^t \perp \mathbf{1}-\zeta^t \geq \mathbf{0}, \\ \mathbf{0}\leq \alpha^t \perp B^t(B^t)^\top\alpha^t - \mathbf{1}+\xi^t \geq \mathbf{0}, \\ \mathbf{0}\leq \xi^t \perp C\mathbf{1}-\alpha^t \geq \mathbf{0}. \end{cases} \end{aligned} \tag{20}$$

The presence of the equilibrium constraints makes problem (20) an instance of an MPEC, which is sometimes labelled as an extension of a bilevel optimization problem Luo et al. (1996). The optimal hyperparameter is now well defined as a global optimal solution to the MPEC Lee et al. (2015). Now, we have transformed a bilevel classification model into an MPEC.

We can also write (20) in a compact form. To proceed, let

$$\zeta := \begin{bmatrix} \zeta^1 \\ \zeta^2 \\ \vdots \\ \zeta^T \end{bmatrix} \in \mathbb{R}^{Tm_1},\ z := \begin{bmatrix} z^1 \\ z^2 \\ \vdots \\ z^T \end{bmatrix} \in \mathbb{R}^{Tm_1},\ A := \begin{bmatrix} A^1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & A^2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & A^T \end{bmatrix} \in \mathbb{R}^{Tm_1\times Tn},$$

and α, ξ, B be defined in (7).



(a) Points with indices in Ψ_1 **(b)** Points with indices in Ψ_2 **(c)** Points with indices in Ψ_3

Fig. 6 Representation of points with index sets Ψ_j, $j = 1, 2, 3$

Then problem (20) can be written as

$$\begin{aligned} \min_{\substack{C\in\mathbb{R} \\ \zeta\in\mathbb{R}^{Tm_1},\ z\in\mathbb{R}^{Tm_1} \\ \alpha\in\mathbb{R}^{Tm_2},\ \xi\in\mathbb{R}^{Tm_2}}} \quad & \frac{1}{Tm_1}\mathbf{1}^\top\zeta \\ \text{s.t. } & \mathbf{0} \le \zeta \perp AB^\top\alpha + z \ge \mathbf{0}, \\ & \mathbf{0} \le z \perp \mathbf{1} - \zeta \ge \mathbf{0}, \\ & \mathbf{0} \le \alpha \perp BB^\top\alpha - \mathbf{1} + \xi \ge \mathbf{0}, \\ & \mathbf{0} \le \xi \perp C\mathbf{1} - \alpha \ge \mathbf{0}. \end{aligned} \tag{21}$$

From now on, all our analysis is going to be based on this model.

3.2 Some properties of the MPEC reformulation

Observe that the last two constraints of problem (21) correspond to the complementarity systems that are part of the KKT conditions of the lower-level problem in (6). As the latter conditions are carefully studied in Proposition 3, it remains to analyze the first two complementarity systems describing the feasible set of problem (21). Hence, we partition them as follows

$$\Psi_1 := \left\{ i \in Q_u \mid 0 \le \zeta_i < 1,\ (AB^\top\alpha + z)_i = 0,\ z_i = 0 \right\}, \tag{22}$$

$$\Psi_2 := \left\{ i \in Q_u \mid \zeta_i = 0,\ (AB^\top\alpha + z)_i > 0,\ z_i = 0 \right\}, \tag{23}$$

$$\Psi_3 := \left\{ i \in Q_u \mid \zeta_i = 1,\ (AB^\top\alpha + z)_i = 0,\ z_i > 0 \right\}. \tag{24}$$

Similarly to (8)–(13), the intersection of any pair of the index sets Ψ_j for $j = 1, 2, 3$ is empty. In the same vein, an illustrative representation of data points corresponding to the index sets Ψ_j for $j = 1, 2, 3$ is given in Fig. 6.

Proposition 4 *Considering the validation points corresponding to Q_u in (1), let (ζ, z, α) satisfy the first two complementarity systems describing the feasible set of problem (21). Then, the following statements hold true:*

(a) *The points* $\{x_i\}_{i\in\Psi_1}$ *lie on the separating hyperplane* H *and are therefore correctly classified.*
(b) *The points* $\{x_i\}_{i\in\Psi_2}$ *lie on the correctly classified side of the separating hyperplane* H *and are therefore correctly classified.*
(c) *The points* $\{x_i\}_{i\in\Psi_3}$ *lie on the misclassified side of the separating hyperplane* H *and are therefore misclassified.*

Proof We take positive points for example. The same analysis can be applied to negative ones. Since $w = B^\top\alpha$ in (5c), we get $(AB^\top\alpha + z)_i = (Aw + z)_i$.

(a) For the points $\{x_i\}_{i\in\Psi_1}$, since $z_i = 0$ in (22), we have $(Aw + z)_i = (Aw)_i = 0$, that is, $y_i(w^\top x_i) = 0$. For a positive point, $y_i = 1$, it implies that $w^\top x_i = 0$. It corresponds to (b) in Proposition 2. Therefore, it means that the point x_i lies on the separating hyperplane H. It is correctly classified.
(b) For the points $\{x_i\}_{i\in\Psi_2}$, since $z_i = 0$ in (23), we have $(Aw + z)_i = (Aw)_i > 0$, that is, $y_i(w^\top x_i) > 0$. For a positive point, $y_i = 1$, it implies that $w^\top x_i > 0$. It corresponds to (c) in Proposition 2. Therefore, it means that the point x_i lies on the correctly classified side of the separating hyperplane H. It is correctly classified.
(c) For the points $\{x_i\}_{i\in\Psi_3}$, since $z_i > 0$ in (24), we have $(Aw)_i < 0$, that is, $y_i(w^\top x_i) < 0$. For a positive point, $y_i = 1$, it implies that $w^\top x_i < 0$. It corresponds to (a) in Proposition 2. Therefore, it means that the point x_i lies on the misclassified side of the separating hyperplane H.

□

In Sect. 4, Proposition 4 will be combined with Proposition 3 to prove Proposition 5. It might also be important to note that if a validation point x_i lies on the separating hyperplane H, then we will have $0 \le \zeta_i < 1$.

3.3 The global relaxation method (GRM)

Here, we present a numerical algorithm to solve the MPEC (21). There are various methods for solving MPECs, we refer to Dempe (2003), Luo et al. (1996) for some surveys on the problem and to Ye (2005), Flegel (2005), Wu et al. (2015), Harder et al. (2021), Guo et al. (2015), Jara-Moroni et al. (2018), Júdice (2012), Li et al. (2015), Yu et al. (2019), Dempe (2003), Anitescu (2000), Facchinei and Pang (2007), Fletcher et al. (2006), Fukushima and Tseng (2002) for some of the latest methods to solve the problem. Among methods to solve MPECs, one of the most popular ones is the relaxation method due to Scholtes (2001). Recently, Hoheisel et al. (2013) provided comparisons of five relaxation methods for solving MPECs, where it appears that the GRM has the best theoretical (in terms of requiring weaker assumptions for convergence) and numerical performance. Therefore, we will apply the GRM to solve our MPEC (21).

To simplify the presentation of the method, we now write problem (21) into a further compact format. Let $v = \left[C,\ \zeta^\top,\ z^\top,\ \alpha^\top,\ \xi^\top\right]^\top \in \mathbb{R}^{\overline{m}+1}$ with $\overline{m} = 2T(m_1 + m_2)$ and define the functions

$$F(v) = M^\top v, \quad G(v) = Pv + a, \quad \text{and} \quad H(v) = Qv, \tag{25}$$

where

$$M=\frac{1}{Tm_1}\begin{bmatrix}0\\ \mathbf{1}_{Tm_1}\\ \mathbf{0}_{Tm_1}\\ \mathbf{0}_{Tm_2}\\ \mathbf{0}_{Tm_2}\end{bmatrix}\in\mathbb{R}^{\overline{m}+1},\ a=\begin{bmatrix}\mathbf{0}_{Tm_1}\\ \mathbf{1}_{Tm_1}\\ -\mathbf{1}_{Tm_2}\\ \mathbf{0}_{Tm_2}\end{bmatrix}\in\mathbb{R}^{\overline{m}},\ Q=\begin{bmatrix}\mathbf{0}_{\overline{m}}\ I_{\overline{m}}\end{bmatrix}\in\mathbb{R}^{\overline{m}\times(\overline{m}+1)},$$

$$P=\begin{bmatrix}\mathbf{0}_{Tm_1} & \mathbf{0}_{Tm_1\times Tm_1} & I_{Tm_1} & AB^{\top} & \mathbf{0}_{Tm_1\times Tm_2}\\ \mathbf{0}_{Tm_1} & -I_{Tm_1} & \mathbf{0}_{Tm_1\times Tm_1} & \mathbf{0}_{Tm_1\times Tm_2} & \mathbf{0}_{Tm_1\times Tm_2}\\ \mathbf{0}_{Tm_2} & \mathbf{0}_{Tm_2\times Tm_1} & \mathbf{0}_{Tm_2\times Tm_1} & BB^{\top} & I_{Tm_2}\\ \mathbf{1}_{Tm_2} & \mathbf{0}_{Tm_2\times Tm_1} & \mathbf{0}_{Tm_2\times Tm_1} & -I_{Tm_2} & \mathbf{0}_{Tm_2\times Tm_2}\end{bmatrix}\in\mathbb{R}^{\overline{m}\times(\overline{m}+1)}.$$

Problem (21) can then be written in the form

$$\begin{aligned}&\min_{v\in\mathbb{R}^{\overline{m}+1}}\ F(v)\\ &\text{s.t. }\ \mathbf{0}\le H(v)\perp G(v)\ge\mathbf{0}.\end{aligned}\tag{26}$$

The basic idea of the GRM is as follows. Let $\{t_k\}\downarrow 0$. At each iteration, we replace the MPEC (26) by the nonlinear program (NLP) of the following form, parameterized in t_k:

$$\begin{aligned}&\min_{v}\ F(v)\\ &\text{s.t. }\ G_i(v)\ge 0\ \ \forall\, i=1,\ \cdots,\ \overline{m},\\ &\quad H_i(v)\ge 0\ \ \forall\, i=1,\ \cdots,\ \overline{m},\\ &\quad G_i(v)H_i(v)\le t_k\ \ \forall\, i=1,\ \cdots,\ \overline{m}.\end{aligned}\tag{NLP-t_k}$$

The details of the GRM are shown in Algorithm 1.

Algorithm 1 The Global Relaxation Method (GRM) $(v_0,\ t_0,\ \sigma,\ t_{min})$

1: **Require** a starting vector v_0, an initial relaxation parameter t_0, and parameters $\sigma\in(0,1)$, $t_{\min}>0$.
2: Set $k:=0$.
3: **while** $t_k>t_{\min}$ **do**
4: Find an approximate solution v^{k+1} of the relaxed problem (NLP-t_k) using v^k as a starting point.
5: Let $t_{k+1}\leftarrow\sigma\cdot t_k$ and $k\leftarrow k+1$.
6: **end while**
7: **Return** the final iterate $v_{opt}:=v^k$, the corresponding function value $F(v_{opt})$, and the maximum constraint violation $\mathrm{Vio}(v_{opt})$.

Here, the maximum violation of all constraints Vio defined by

$$\mathrm{Vio}\ \left(v_{opt}\right)=\|\min\{G(v_{opt}),\ H(v_{opt})\}\|_{\infty}\tag{27}$$

is used to measure the feasibility of the final iterate v_{opt}, where $\|\cdot\|_\infty$ denotes the l_∞ norm. Note that in step 4, the approximate solution refers to the approximate stationary point, in the sense that it satisfies the KKT conditions of (NLP-t_k) approximately. Numerically, we use the SNOPT solver Gill et al. (2002) to compute the KKT points of (NLP-t_k) approximately, such that the norm of the KKT conditions is less than a threshold value $\epsilon = 10^{-6}$. The point v^{k+1} returned by the SNOPT solver is referred to as an approximate solution of (NLP-t_k). We use the GRM in Algorithm 1 to solve the MPEC (26), and get the optimal hyperparameter C and the corresponding function value $F(v_{opt})$ which is the cross-validation error (CV error) measured on the validation sets in T-fold cross-validation. To analyze the convergence of the GRM, we need the concept of C-stationarity, which we define next.

To proceed, let v be a feasible point for the MPEC (26) and recall that $F(v)$, $G(v)$ and $H(v)$ are defined in (25). Based on v, let

$$\begin{aligned} I_G &:= \{i \mid G_i(v) = 0,\ H_i(v) > 0\}, \\ I_{GH} &:= \{i \mid G_i(v) = 0,\ H_i(v) = 0\}, \\ I_H &:= \{i \mid G_i(v) > 0,\ H_i(v) = 0\}. \end{aligned}$$

Definition 1 (C-stationarity) Let v be a feasible point for the MPEC (26). Then v is said to be a C-stationary point, if there are multipliers $\gamma,\ \nu \in \mathbb{R}^{\overline{m}}$, such that

$$\nabla F(v) - \sum_{i=1}^{\overline{m}} \gamma_i \nabla G_i(v) - \sum_{i=1}^{\overline{m}} \nu_i \nabla H_i(v) = \mathbf{0},$$

and $\gamma_i = 0$ for $i \in I_H$, $\nu_i = 0$ for $i \in I_G$, and $\gamma_i \nu_i \geq 0$ for $i \in I_{GH}$.

Note that for problem (26), C-stationarity holds at any local optimal solution that satisfies the MPEC–MFCQ, which can be defined as follows Hoheisel et al. (2013).

Definition 2 A feasible point v for problem (26) satisfies the MPEC-MFCQ if and only if the set of gradient vectors

$$\{\nabla G_i(v) \mid i \in I_G \cup I_{GH}\} \cup \{\nabla H_i(v) \mid i \in I_H \cup I_{GH}\} \tag{28}$$

is positive-linearly independent.

Recall that the set of gradient vectors in (28) is said to be positive-linearly *dependent* if there exist scalars $\{\delta_i\}_{i \in I_G \cup I_{GH}}$ and $\{\beta_i\}_{i \in I_H \cup I_{GH}}$ with $\delta_i \geq 0$ for $i \in I_G \cup I_{GH}$, $\beta_i \geq 0$ for $i \in I_H \cup I_{GH}$, not all of them being zero, such that $\Sigma_{i \in I_G \cup I_{GH}} \delta_i \nabla G_i(v) + \Sigma_{i \in I_H \cup I_{GH}} \beta_i \nabla H_i(v) = \mathbf{0}$. Otherwise, we say that this set of gradient vectors is positive-linearly *independent*.

Also note that various other stationarity concepts can be defined for problem (26); for more details on this, interested readers are referred to Dempe and Zemkoho (2012), Flegel (2005).

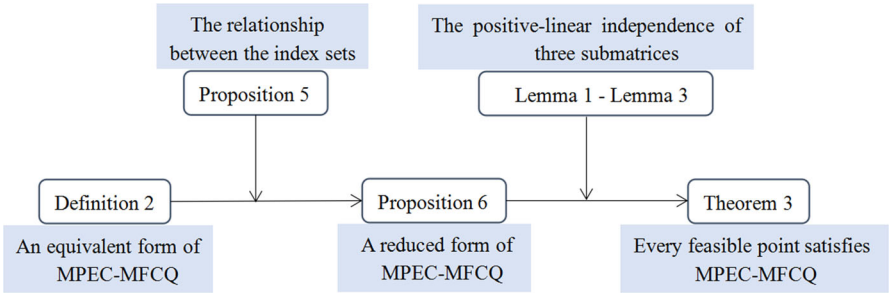


Fig. 7 The roadmap of the proof of the MPEC–MFCQ

The following result establishes that Algorithm 1 is well-defined, as it provides a framework ensuring that a solution (or a stationary point, to be precise) exists for problem (NLP-t_k) as required.

Theorem 1 Hoheisel et al. (2013) *Let v be a feasible point for the MPEC (26) such that MPEC-MFCQ is satisfied at v. Then there exists a neighborhood N of v and $\bar{t} > 0$ such that standard MFCQ for (NLP-t_k) at $t_k = t$ is satisfied at all feasible points of (NLP-t_k) at $t_k = t$ in this neighborhood N for all $t \in (0, \bar{t})$.*

Subsequently, we have the following convergence result, which ensures that a sequence of stationary points of problem (NLP-t_k), computed by Algorithm 1, converges to a C-stationary point of problem (26).

Theorem 2 Hoheisel et al. (2013) *Let $\{t_k\} \downarrow 0$ and let v^k be a stationary point of (NLP-t_k) with $v^k \to v$ such that MPEC-MFCQ holds at the feasible point v. Then v is a C-stationary point of the MPEC (26).*

Clearly, the MPEC-MFCQ is crucial for the analysis of problem (26), as it not only ensures that the C-stationarity condition can hold at a locally optimal point, but also helps in establishing the two fundamental results in Theorems 1 and 2. Considering this importance of the condition, we carefully analyze it in the next section, and show, in particular, that it automatically holds at any feasible point of problem (26).

4 Fulfilment of the MPEC–MFCQ

In this section, we prove that every point in the feasible set of the MPEC (26) satisfies the MPEC–MFCQ. The rough idea of our proof is as follows. Firstly, by analyzing the relationship of different index sets (Proposition 5), we reach a reduced form of the MPEC–MFCQ (Proposition 6). Then based on the positive-linear independence of three submatrices (Lemmas 1–3), we eventually show the MPEC–MFCQ in Theorem 3. The roadmap of the proof is summarized in Fig. 7.

4.1 Relationships between the index sets

In this part, we first explore more properties about the index sets $I_H,\ I_G,\ I_{GH}$, as they are the key to the analysis of the positive-linear independence of the vectors in (28). Let $I_H := \overset{4}{\underset{k=1}{\cup}} I_{H_k},\ I_G := \overset{4}{\underset{k=1}{\cup}} I_{G_k}$, and $I_{GH} := \overset{4}{\underset{k=1}{\cup}} I_{GH_k}$, where

$$
\begin{aligned}
I_{H_1} &:= \{i \in Q_u \mid \zeta_i = 0,\ (AB^\top\alpha + z)_i > 0\}, && \text{(29a)}\\
I_{H_2} &:= \{i \in Q_u \mid z_i = 0,\ 1 - \zeta_i > 0\}, && \text{(29b)}\\
I_{H_3} &:= \{i \in Q_l \mid \alpha_i = 0,\ (BB^\top\alpha - \mathbf{1} + \xi)_i > 0\}, && \text{(29c)}\\
I_{H_4} &:= \{i \in Q_l \mid \xi_i = 0,\ C - \alpha_i > 0\}, && \text{(29d)}\\
I_{G_1} &:= \{i \in Q_u \mid \zeta_i > 0,\ (AB^\top\alpha + z)_i = 0\}, && \text{(29e)}\\
I_{G_2} &:= \{i \in Q_u \mid z_i > 0,\ 1 - \zeta_i = 0\}, && \text{(29f)}\\
I_{G_3} &:= \{i \in Q_l \mid \alpha_i > 0,\ (BB^\top\alpha - \mathbf{1} + \xi)_i = 0\}, && \text{(29g)}\\
I_{G_4} &:= \{i \in Q_l \mid \xi_i > 0,\ C - \alpha_i = 0\}, && \text{(29h)}\\
I_{GH_1} &:= \{i \in Q_u \mid \zeta_i = 0,\ (AB^\top\alpha + z)_i = 0\}, && \text{(29i)}\\
I_{GH_2} &:= \{i \in Q_u \mid z_i = 0,\ 1 - \zeta_i = 0\}, && \text{(29j)}\\
I_{GH_3} &:= \{i \in Q_l \mid \alpha_i = 0,\ (BB^\top\alpha - \mathbf{1} + \xi)_i = 0\}, && \text{(29k)}\\
I_{GH_4} &:= \{i \in Q_l \mid \xi_i = 0,\ C - \alpha_i = 0\}. && \text{(29l)}
\end{aligned}
$$

Here, $Q_u,\ Q_l$ are defined in (1) and (2), respectively. Furthermore, let

$$I^k := I_{H_k} \cup I_{G_k} \cup I_{GH_k},\ k = 1,\ 2,\ 3,\ 4.$$

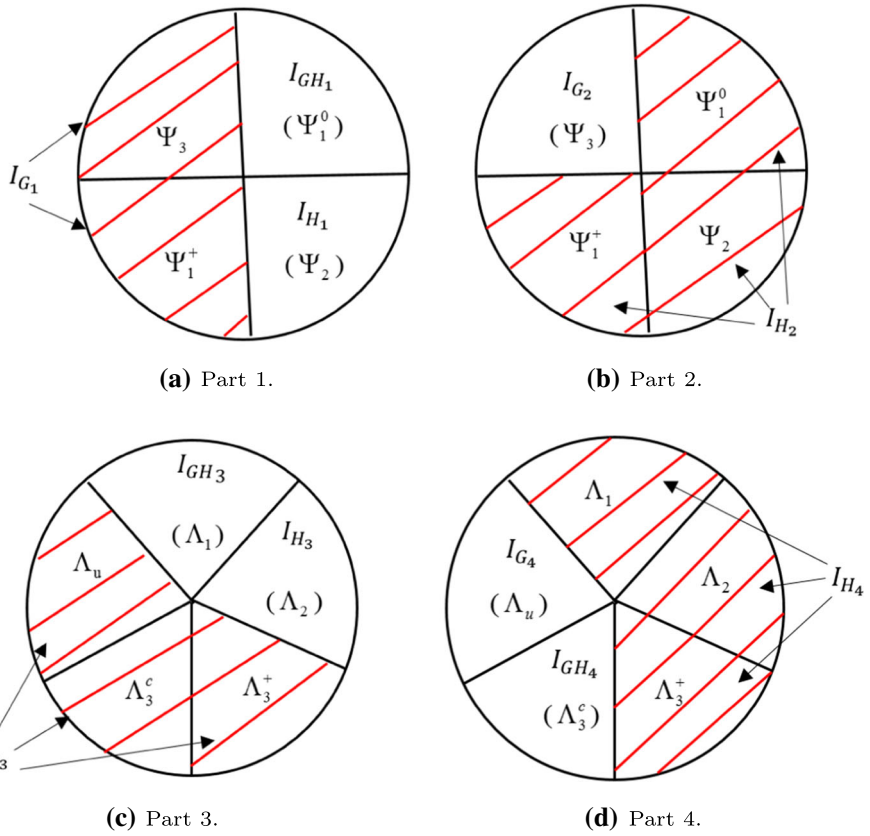
It can be observed that each index set $I^k,\ k = 1,\ 2,\ 3,\ 4$ corresponds to the union of the three components in the partition involved in the corresponding part of the complementarity systems in (21); that is,

Part 1: I^1 for the partition of the system $\mathbf{0} \le \zeta \perp AB^T\alpha + z \ge \mathbf{0}$;
Part 2: I^2 for the partition of the system $\mathbf{0} \le z \perp \mathbf{1} - \zeta \ge \mathbf{0}$;
Part 3: I^3 for the partition of the system $\mathbf{0} \le \alpha \perp BB^T\alpha - \mathbf{1} + \xi \ge \mathbf{0}$;
Part 4: I^4 for the partition of the system $\mathbf{0} \le \xi \perp C\mathbf{1} - \alpha \ge \mathbf{0}$.

In the previous section, we have clarified a one-to-one correspondence between the index set of the validation points Q_u in (1) and the complementary constraints in Part 1 and Part 2, respectively. It is clearly that $I^1 = I^2 = Q_u$. Similarly, we have $I^3 = I^4 = Q_l$.

Next, we give the relationships between the index sets in (29); recall that we already have some index sets described in Propositions 3 and 4. For the convenience of the analysis, we divide the index set Λ_3 in (10) into two subsets Λ_3^+ and Λ_3^c, as well as Ψ_1 in (22) into Ψ_1^0 and Ψ_1^+:

$$\Lambda_3^+ := \{i \in Q_l \mid 0 < \alpha_i < C,\ (BB^\top\alpha - \mathbf{1} + \xi)_i = 0,\ \xi_i = 0\}, \qquad (30)$$



(a) Part 1. **(b)** Part 2.

(c) Part 3. **(d)** Part 4.

Fig. 8 The index sets corresponding to the complementarity constraints in Parts 1–4

$$\Lambda_3^c := \{i \in Q_l \mid \alpha_i = C,\ (BB^\top\alpha - \mathbf{1} + \xi)_i = 0,\ \xi_i = 0\}, \tag{31}$$

$$\Psi_1^0 := \{i \in Q_u \mid \zeta_i = 0,\ (AB^\top\alpha + z)_i = 0,\ z_i = 0\}, \tag{32}$$

$$\Psi_1^+ := \{i \in Q_u \mid 0 < \zeta_i < 1,\ (AB^\top\alpha + z)_i = 0,\ z_i = 0\}. \tag{33}$$

Proposition 5 *The index sets in (29) and the index sets in Proposition 3 and Proposition 4 have the following relationship:*

(a) *In Part* 1, $I_{H_1} = \Psi_2,\ I_{G_1} = \Psi_1^+ \cup \Psi_3,\ I_{GH_1} = \Psi_1^0$.
(b) *In Part* 2, $I_{H_2} = \Psi_1 \cup \Psi_2,\ I_{G_2} = \Psi_3,\ I_{GH_2} = \emptyset$.
(c) *In Part* 3, $I_{H_3} = \Lambda_2,\ I_{G_3} = \Lambda_3 \cup \Lambda_u,\ I_{GH_3} = \Lambda_1$.
(d) *In Part* 4, $I_{H_4} = \Lambda_1 \cup \Lambda_2 \cup \Lambda_3^+,\ I_{G_4} = \Lambda_u,\ I_{GH_4} = \Lambda_3^c$.

Here, Λ_u *is defined as follows*

$$\Lambda_u := \left\{i \in Q_l \mid \alpha_i = C,\ (BB^\top\alpha - \mathbf{1} + \xi)_i = 0,\ \xi_i > 0\right\} = \Lambda_4 \cup \Lambda_5 \cup \Lambda_6. \tag{34}$$

Proof According to the definition of the index sets in (29) and the index sets in Proposition 3 and Proposition 4, we have the following analysis.

(a) In Part 1, for $i \in I_{H_1}$, compared with the index set Ψ_2 in (23), it follows that we have $z_i = 0$ and $I_{H_1} = \Psi_2$. For $i \in I_{G_1}$, compared with the index sets Ψ_1^+ in (33) and Ψ_3 in (24), we get $0 < \zeta_i < 1$, $z_i = 0$ or $\zeta_i = 1$, $z_i > 0$, and $I_{G_1} = \Psi_1^+ \cup \Psi_3$. For $i \in I_{GH_1}$, compared with the index set Ψ_1^0 in (32), we get $z_i = 0$ and $I_{GH_1} = \Psi_1^0$.

(b) In Part 2, for $i \in I_{H_2}$, compared with the index sets Ψ_1 in (22) and Ψ_2 in (23), we get $I_{H_2} = \Psi_1 \cup \Psi_2$. For $i \in I_{G_2}$, compared with the index set Ψ_3 in (24), we get $(AB^\top \alpha + z)_i = 0$ and $I_{G_2} = \Psi_3$. For $i \in I_{GH_2}$, there is no index set in Proposition 4 corresponds to the index set I_{GH_2}. Therefore, $I_{GH_2} = \emptyset$.

(c) In Part 3, for $i \in I_{H_3}$, compared with the index set Λ_2 in (9), we get $\xi_i = 0$ and $I_{H_3} = \Lambda_2$. For $i \in I_{G_3}$, compared with the index sets Λ_3 in (10) and Λ_u in (34), we get $I_{G_3} = \Lambda_3 \cup \Lambda_u$. For $i \in I_{GH_3}$, compared with the index set Λ_1 in (8), we get $\xi_i = 0$ and $I_{GH_3} = \Lambda_1$.

(d) In Part 4, for $i \in I_{H_4}$, compared with the index sets Λ_1 in (8), Λ_2 in (9) and Λ_3^+ in (30), we get $I_{H_4} = \Lambda_1 \cup \Lambda_2 \cup \Lambda_3^+$. For $i \in I_{G_4}$, compared with the index set Λ_u in (34), we get $(Bw - \mathbf{1} + \xi)_i = 0$ and $I_{G_4} = \Lambda_u$. For $i \in I_{GH_4}$, compared with the index set Λ_3^c in (31), it results that we have $(Bw - \mathbf{1} + \xi)_i = 0$ and $I_{GH_4} = \Lambda_3^c$.

The results in Proposition 5 are demonstrated in Fig. 8. For example, for (a) in Proposition 5, the index sets of complementarity constraints in Part 1 are shown in Fig. 8a, which is about the relationship of I_{H_1}, I_{G_1}, I_{GH_1} in (29) and the index sets (22)–(24). In Fig. 8a, the red shaded part represents the index set I_{G_1}, which contains the index sets Ψ_1^+ and Ψ_3. (b)–(d) in Proposition 5 are demonstrated in Fig. 8 b–d. Specially, in Fig. 8b, the red shaded part represents the index set I_{H_2}, which contains the index sets Ψ_1 (or $\Psi_1^0 \cup \Psi_1^+$) and Ψ_2. In Fig. 8c, the red shaded part represents the index set I_{G_3}, which contains the index sets Λ_3 (or $\Lambda_3^+ \cup \Lambda_3^c$) and Λ_u. In Fig. 8d, the red shaded part represents the index set I_{H_4}, which contains the index sets Λ_1, Λ_2, and Λ_3^+.

4.2 The reduced form of the MPEC-MFCQ

Proposition 6 *The set of gradient vectors in* (28) *at a feasible point v for the MPEC* (26) *can be written in the matrix form*

$$\Gamma = \begin{bmatrix}
\mathbf{0}_{(I_{G_1},\, L_1)} & \mathbf{0}_{(I_{G_1},\, L_2)} & \Gamma^3_a & (AB^\top)_{(I_{G_1},\, \cdot)} & \mathbf{0}_{(I_{G_1},\, L_5)} \\
\mathbf{0}_{(I_{GH_1},\, L_1)} & \mathbf{0}_{(I_{GH_1},\, L_2)} & \Gamma^3_b & (AB^\top)_{(I_{GH_1},\, \cdot)} & \mathbf{0}_{(I_{GH_1},\, L_5)} \\
\mathbf{0}_{(I_{GH_1},\, L_1)} & \Gamma^2_c & \mathbf{0}_{(I_{GH_1},\, L_3)} & \mathbf{0}_{(I_{GH_1},\, L_4)} & \mathbf{0}_{(I_{GH_1},\, L_5)} \\
\mathbf{0}_{(I_{H_1},\, L_1)} & \Gamma^2_d & \mathbf{0}_{(I_{H_1},\, L_3)} & \mathbf{0}_{(I_{H_1},\, L_4)} & \mathbf{0}_{(I_{H_1},\, L_5)} \\
\mathbf{0}_{(I_{G_2},\, L_1)} & \Gamma^2_e & \mathbf{0}_{(I_{G_2},\, L_3)} & \mathbf{0}_{(I_{G_2},\, L_4)} & \mathbf{0}_{(I_{G_2},\, L_5)} \\
\mathbf{0}_{(I_{H_2},\, L_1)} & \mathbf{0}_{(I_{H_2},\, L_2)} & \Gamma^3_f & \mathbf{0}_{(I_{H_2},\, L_4)} & \mathbf{0}_{(I_{H_2},\, L_5)} \\
\mathbf{0}_{(I_{G_3},\, L_1)} & \mathbf{0}_{(I_{G_3},\, L_2)} & \mathbf{0}_{(I_{G_3},\, L_3)} & (BB^\top)_{(I_{G_3},\, \cdot)} & \Gamma^5_g \\
\mathbf{0}_{(I_{GH_3},\, L_1)} & \mathbf{0}_{(I_{GH_3},\, L_2)} & \mathbf{0}_{(I_{GH_3},\, L_3)} & (BB^\top)_{(I_{GH_3},\, \cdot)} & \Gamma^5_h \\
\mathbf{0}_{(I_{GH_3},\, L_1)} & \mathbf{0}_{(I_{GH_3},\, L_2)} & \mathbf{0}_{(I_{GH_3},\, L_3)} & \Gamma^4_i & \mathbf{0}_{(I_{GH_3},\, L_5)} \\
\mathbf{0}_{(I_{H_3},\, L_1)} & \mathbf{0}_{(I_{H_3},\, L_2)} & \mathbf{0}_{(I_{H_3},\, L_3)} & \Gamma^4_j & \mathbf{0}_{(I_{H_3},\, L_5)} \\
\mathbf{1}_{(I_{G_4},\, L_1)} & \mathbf{0}_{(I_{G_4},\, L_2)} & \mathbf{0}_{(I_{G_4},\, L_3)} & \Gamma^4_k & \mathbf{0}_{(I_{G_4},\, L_5)} \\
\mathbf{1}_{(I_{GH_4},\, L_1)} & \mathbf{0}_{(I_{GH_4},\, L_2)} & \mathbf{0}_{(I_{GH_4},\, L_3)} & \Gamma^4_l & \mathbf{0}_{(I_{GH_4},\, L_5)} \\
\mathbf{0}_{(I_{GH_4},\, L_1)} & \mathbf{0}_{(I_{GH_4},\, L_2)} & \mathbf{0}_{(I_{GH_4},\, L_3)} & \mathbf{0}_{(I_{GH_4},\, L_4)} & \Gamma^5_m \\
\mathbf{0}_{(I_{H_4},\, L_1)} & \mathbf{0}_{(I_{H_4},\, L_2)} & \mathbf{0}_{(I_{H_4},\, L_3)} & \mathbf{0}_{(I_{H_4},\, L_4)} & \Gamma^5_n
\end{bmatrix}, \tag{35}$$

where $L_q,\ q = 1, \cdots, 5$ *are the index sets of columns corresponding to the variables* C, ζ, z, α, *and* ξ, *respectively, and*

$$\left.\begin{aligned}
\Gamma^3_a &:= \left[\mathbf{0}_{(I_{G_1},\, \Psi_1^0 \cup \Psi_2)}\ I_{(I_{G_1},\, \Psi_1^+ \cup \Psi_3)}\right] \\
\Gamma^3_b &:= \left[I_{(I_{GH_1},\, \Psi_1^0)}\ \mathbf{0}_{(I_{GH_1},\, \Psi_1^+ \cup \Psi_2 \cup \Psi_3)}\right] \\
\Gamma^2_c &:= \left[I_{(I_{GH_1},\, \Psi_1^0)}\ \mathbf{0}_{(I_{GH_1},\, \Psi_1^+ \cup \Psi_2 \cup \Psi_3)}\right] \\
\Gamma^2_d &:= \left[\mathbf{0}_{(I_{H_1},\, \Psi_1 \cup \Psi_3)}\ I_{(I_{H_1},\, \Psi_2)}\right] \\
\Gamma^2_e &:= \left[\mathbf{0}_{(I_{G_2},\, \Psi_1 \cup \Psi_2)}\ -I_{(I_{G_2},\, \Psi_3)}\right] \\
\Gamma^3_f &:= \left[I_{(I_{H_2},\, \Psi_1 \cup \Psi_2)}\ \mathbf{0}_{(I_{H_2},\, \Psi_3)}\right] \\
\Gamma^5_g &:= \left[\mathbf{0}_{(I_{G_3},\, \Lambda_1 \cup \Lambda_2)}\ I_{(I_{G_3},\, \Lambda_3 \cup \Lambda_u)}\right] \\
\Gamma^5_h &:= \left[I_{(I_{GH_3},\, \Lambda_1)}\ \mathbf{0}_{(I_{GH_3},\, \Lambda_2 \cup \Lambda_3 \cup \Lambda_u)}\right] \\
\Gamma^4_i &:= \left[I_{(I_{GH_3},\, \Lambda_1)}\ \mathbf{0}_{(I_{GH_3},\, \Lambda_2 \cup \Lambda_3 \cup \Lambda_u)}\right] \\
\Gamma^4_j &:= \left[\mathbf{0}_{(I_{H_3},\, \Lambda_1 \cup \Lambda_3 \cup \Lambda_u)}\ I_{(I_{H_3},\, \Lambda_2)}\right] \\
\Gamma^4_k &:= \left[\mathbf{0}_{(I_{G_4},\, \Lambda_1 \cup \Lambda_2 \cup \Lambda_3)}\ -I_{(I_{G_4},\, \Lambda_u)}\right] \\
\Gamma^4_l &:= \left[\mathbf{0}_{(I_{GH_4},\, \Lambda_1 \cup \Lambda_2 \cup \Lambda_3^+ \cup \Lambda_u)}\ -I_{(I_{GH_4},\, \Lambda_3^c)}\right] \\
\Gamma^5_m &:= \left[\mathbf{0}_{(I_{GH_4},\, \Lambda_1 \cup \Lambda_2 \cup \Lambda_3^+ \cup \Lambda_u)}\ I_{(I_{GH_4},\, \Lambda_3^c)}\right] \\
\Gamma^5_n &:= \left[I_{(I_{H_4},\, \Lambda_1 \cup \Lambda_2 \cup \Lambda_3^+)}\ \mathbf{0}_{(I_{H_4},\, \Lambda_3^c \cup \Lambda_u)}\right]
\end{aligned}\right\}. \tag{36}$$

Proof Based on Definition 2, we can write the set of gradient vectors in (28) at a feasible point v in the rows of the matrix Γ as follows

$$\Gamma = \begin{bmatrix} \nabla G(v)_{I_{G_1}} \\ \nabla G(v)_{I_{GH_1}} \\ \nabla H(v)_{I_{GH_1}} \\ \nabla H(v)_{I_{H_1}} \\ \nabla G(v)_{I_{G_2}} \\ \nabla H(v)_{I_{H_2}} \\ \nabla G(v)_{I_{G_3}} \\ \nabla G(v)_{I_{GH_3}} \\ \nabla H(v)_{I_{GH_3}} \\ \nabla H(v)_{I_{H_3}} \\ \nabla G(v)_{I_{G_4}} \\ \nabla G(v)_{I_{GH_4}} \\ \nabla H(v)_{I_{GH_4}} \\ \nabla H(v)_{I_{H_4}} \end{bmatrix}. \tag{37}$$

Now, we can easily show that the matrix Γ in (37) is equivalent to the more specific form in (35). To proceed, first note that from Proposition 5 (a) and (b), we have

$$I_{H_1} = \Psi_2,\ I_{G_1} = \Psi_1^+ \cup \Psi_3,\ I_{GH_1} = \Psi_1^0,\ I_{H_2} = \Psi_1 \cup \Psi_2,$$
$$I_{G_2} = \Psi_3,\ Q_u = \Psi_1 \cup \Psi_2 \cup \Psi_3.$$

So, we get Γ_a^3, Γ_b^3, Γ_c^2, Γ_d^2, Γ_e^2, and Γ_f^3 in (36). On the other hand, it follows from Proposition 5 (c) and (d), we have

$$I_{H_3} = \Lambda_2,\ I_{G_3} = \Lambda_3 \cup \Lambda_u,\ I_{GH_3} = \Lambda_1,\ I_{H_4} = \Lambda_1 \cup \Lambda_2 \cup \Lambda_3^+,$$
$$I_{G_4} = \Lambda_u,\ I_{GH_4} = \Lambda_3^c,\ \ Q_l = \Lambda_1 \cup \Lambda_2 \cup \Lambda_3 \cup \Lambda_u.$$

Subsequently, it follows that Γ_g^5, Γ_h^5, Γ_i^4, Γ_j^4, Γ_k^4, Γ_l^4, Γ_m^5, and Γ_n^5 in (36). Therefore, we obtain the form of the matrix Γ in (35). □

4.3 Three important lemmas

Due to the complicated form of Γ in (35), in this part, we first present three lemmas, addressing the positive-linear independence of three submatrices in Γ marked by blue, green and yellow, respectively. To proceed from here on, we define the size of each index set in (29) and Propositions 3–4 as follows. We denote the size of the index set I_{G_1} by S_1, that is, $|I_{G_1}| = S_1$. Similarly,

$$|I_{G_2}| = S_2, \quad |I_{G_3}| = S_3, \quad |I_{G_4}| = S_4,$$
$$|I_{H_1}| = U_1, \quad |I_{H_2}| = U_2, \quad |I_{H_3}| = U_3, \quad |I_{H_4}| = U_4,$$
$$|I_{GH_1}| = W_1, \ |I_{GH_3}| = W_2, \ |I_{GH_4}| = W_3,$$
$$|\Lambda_1| = D_1, \quad |\Lambda_2| = D_2, \quad |\Lambda_3^+| = D_3, \quad |\Lambda_3^c| = D_4, \ |\Lambda_u| = D_5,$$
$$|\Psi_1^0| = N_1, \quad |\Psi_1^+| = N_2, \quad |\Psi_2| = N_3, \quad |\Psi_3| = N_4.$$

Further, we denote the index corresponding to each row in the matrices $\Gamma_a^3, \cdots \Gamma_n^5$, in (36) by $a_s, \cdots n_s$, respectively.

Lemma 1 *The row vectors in the following matrix*

$$\begin{bmatrix} \Gamma_c^2 \\ \Gamma_d^2 \\ \Gamma_e^2 \end{bmatrix} = \begin{bmatrix} I_{(IGH_1,\ \Psi_1^0)} & \mathbf{0}_{(IGH_1,\ \Psi_1^+)} & \mathbf{0}_{(IGH_1,\ \Psi_2)} & \mathbf{0}_{(IGH_1,\ \Psi_3)} \\ \mathbf{0}_{(IH_1,\ \Psi_1^0)} & \mathbf{0}_{(IH_1,\ \Psi_1^+)} & I_{(IH_1,\ \Psi_2)} & \mathbf{0}_{(IH_1,\ \Psi_3)} \\ \mathbf{0}_{(IG_2,\ \Psi_1^0)} & \mathbf{0}_{(IG_2,\ \Psi_1^+)} & \mathbf{0}_{(IG_2,\ \Psi_2)} & -I_{(IG_2,\ \Psi_3)} \end{bmatrix} \tag{38}$$

are positive-linearly independent.

Proof Assume that there exist $\overline{\rho}^c \in \mathbb{R}^{W_1}$ and $\overline{\rho}^c \geq \mathbf{0}$, $\overline{\rho}^d \in \mathbb{R}^{U_1}$ and $\overline{\rho}^d \geq \mathbf{0}$, $\overline{\rho}^e \in \mathbb{R}^{S_2}$ and $\overline{\rho}^e \geq \mathbf{0}$, such that

$$\sum_{s=1}^{W_1} \rho_s^c \begin{bmatrix} e_{c_s}^{W_1} \\ \mathbf{0}_{N_2} \\ \mathbf{0}_{N_3} \\ \mathbf{0}_{N_4} \end{bmatrix} + \sum_{s=1}^{U_1} \rho_s^d \begin{bmatrix} \mathbf{0}_{N_1} \\ \mathbf{0}_{N_2} \\ e_{d_s}^{U_1} \\ \mathbf{0}_{N_4} \end{bmatrix} + \sum_{s=1}^{S_2} \rho_s^e \begin{bmatrix} \mathbf{0}_{N_1} \\ \mathbf{0}_{N_2} \\ \mathbf{0}_{N_3} \\ -e_{e_s}^{S_2} \end{bmatrix} = \mathbf{0}.$$

The above equation is equivalent to the following system

$$\begin{bmatrix} \overline{\rho}^c \\ \mathbf{0}_{N_2} \\ \overline{\rho}^d \\ -\overline{\rho}^e \end{bmatrix} = \mathbf{0}. \tag{39}$$

Since $\overline{\rho}^c \geq \mathbf{0}$, $\overline{\rho}^d \geq \mathbf{0}$, $\overline{\rho}^e \geq \mathbf{0}$, we get $\overline{\rho}^c = \mathbf{0}$, $\overline{\rho}^d = \mathbf{0}$, $\overline{\rho}^e = \mathbf{0}$ from Eq. (39). Therefore, the row vectors in the matrix (38) are positive-linearly independent. □

Lemma 2 *The row vectors in the following matrix*

$$\begin{bmatrix} \Gamma_a^3 \\ \Gamma_b^3 \\ \Gamma_f^3 \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{(\Psi_1^+,\ \Psi_1^0)} & I_{(\Psi_1^+,\ \Psi_1^+)} & \mathbf{0}_{(\Psi_1^+,\ \Psi_2)} & \mathbf{0}_{(\Psi_1^+,\ \Psi_3)} \\ \mathbf{0}_{(\Psi_3,\ \Psi_1^0)} & \mathbf{0}_{(\Psi_3,\ \Psi_1^+)} & \mathbf{0}_{(\Psi_3,\ \Psi_2)} & I_{(\Psi_3,\ \Psi_3)} \\ I_{(IGH_1,\ \Psi_1^0)} & \mathbf{0}_{(IGH_1,\ \Psi_1^+)} & \mathbf{0}_{(IGH_1,\ \Psi_2)} & \mathbf{0}_{(IGH_1,\ \Psi_3)} \\ I_{(\Psi_1^0,\ \Psi_1^0)} & \mathbf{0}_{(\Psi_1^0,\ \Psi_1^+)} & \mathbf{0}_{(\Psi_1^0,\ \Psi_2)} & \mathbf{0}_{(\Psi_1^0,\ \Psi_3)} \\ \mathbf{0}_{(\Psi_1^+,\ \Psi_1^0)} & I_{(\Psi_1^+,\ \Psi_1^+)} & \mathbf{0}_{(\Psi_1^+,\ \Psi_2)} & \mathbf{0}_{(\Psi_1^+,\ \Psi_3)} \\ \mathbf{0}_{(\Psi_2,\ \Psi_1^0)} & \mathbf{0}_{(\Psi_2,\ \Psi_1^+)} & I_{(\Psi_2,\ \Psi_2)} & \mathbf{0}_{(\Psi_2,\ \Psi_3)} \end{bmatrix} \tag{40}$$

are positive-linearly independent.

Proof Assume that there exist $\overline{\rho}^a \in \mathbb{R}^{S_1}$ and $\overline{\rho}^a \geq \mathbf{0}$, $\overline{\rho}^b \in \mathbb{R}^{W_1}$ and $\overline{\rho}^b \geq \mathbf{0}$, $\overline{\rho}^f \in \mathbb{R}^{U_2}$ and $\overline{\rho}^f \geq \mathbf{0}$, such that

$$\sum_{s=1}^{S_1} \rho_s^a \begin{bmatrix} \mathbf{0}_{N_1+N_3} \\ e_{a_s}^{S_1} \end{bmatrix} + \sum_{s=1}^{W_1} \rho_s^b \begin{bmatrix} e_{b_s}^{W_1} \\ \mathbf{0}_{N_2+N_3+N_4} \end{bmatrix} + \sum_{s=1}^{U_2} \rho_s^f \begin{bmatrix} e_{f_s}^{U_2} \\ \mathbf{0}_{N_4} \end{bmatrix} = \mathbf{0}.$$

The above equation is equivalent to the following system

$$\begin{bmatrix} \overline{\rho}^b + \overline{\rho}^f_{\Psi_1^0} \\ \overline{\rho}^a_{\Psi_1^+} + \overline{\rho}^f_{\Psi_1^+} \\ \overline{\rho}^f_{\Psi_2} \\ \overline{\rho}^a_{\Psi_3} \end{bmatrix} = \mathbf{0}. \tag{41}$$

Since $\overline{\rho}^a \geq \mathbf{0},\ \overline{\rho}^b \geq \mathbf{0},\ \overline{\rho}^f \geq \mathbf{0}$, we get $\overline{\rho}^a = \mathbf{0},\ \overline{\rho}^b = \mathbf{0},\ \overline{\rho}^f = \mathbf{0}$ from Eq. (41). Therefore, the row vectors in the matrix (40) are positive-linearly independent. □

Lemma 3 *The row vectors in the matrix* Γ_{sub} *defined by*

$$\Gamma_{sub} = \begin{bmatrix} (BB^\top)_{(I_{G_3},\,\cdot)} & \Gamma_g^5 \\ (BB^\top)_{(I_{GH_3},\,\cdot)} & \Gamma_h^5 \\ \Gamma_i^4 & \mathbf{0}_{(I_{GH_3},\,L_5)} \\ \Gamma_j^4 & \mathbf{0}_{(I_{H_3},\,L_5)} \\ \mathbf{0}_{(I_{GH_4},\,L_4)} & \Gamma_m^5 \\ \mathbf{0}_{(I_{H_4},\,L_4)} & \Gamma_n^5 \end{bmatrix} \tag{42}$$

are positive-linearly independent.

Proof For the convenience of analysis, note that

$$\begin{bmatrix} \Gamma_g^5 \\ \Gamma_h^5 \\ \Gamma_m^5 \\ \Gamma_n^5 \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{(\Lambda_3^+,\,\Lambda_1)} & \mathbf{0}_{(\Lambda_3^+,\,\Lambda_2)} & I_{(\Lambda_3^+,\,\Lambda_3^+)} & \mathbf{0}_{(\Lambda_3^+,\,\Lambda_3^c)} & \mathbf{0}_{(\Lambda_3^+,\,\Lambda_u)} \\ \mathbf{0}_{(\Lambda_3^c,\,\Lambda_1)} & \mathbf{0}_{(\Lambda_3^c,\,\Lambda_2)} & \mathbf{0}_{(\Lambda_3^c,\,\Lambda_3^+)} & I_{(\Lambda_3^c,\,\Lambda_3^c)} & \mathbf{0}_{(\Lambda_3^c,\,\Lambda_u)} \\ \mathbf{0}_{(\Lambda_u,\,\Lambda_1)} & \mathbf{0}_{(\Lambda_u,\,\Lambda_2)} & \mathbf{0}_{(\Lambda_u,\,\Lambda_3^+)} & \mathbf{0}_{(\Lambda_u,\,\Lambda_3^c)} & I_{(\Lambda_u,\,\Lambda_u)} \\ I_{(I_{GH_3},\,\Lambda_1)} & \mathbf{0}_{(I_{GH_3},\,\Lambda_2)} & \mathbf{0}_{(I_{GH_3},\,\Lambda_3^+)} & \mathbf{0}_{(I_{GH_3},\,\Lambda_3^c)} & \mathbf{0}_{(I_{GH_3},\,\Lambda_u)} \\ \mathbf{0}_{(I_{GH_4},\,\Lambda_1)} & \mathbf{0}_{(I_{GH_4},\,\Lambda_2)} & \mathbf{0}_{(I_{GH_4},\,\Lambda_3^+)} & I_{(I_{GH_4},\,\Lambda_3^c)} & \mathbf{0}_{(I_{GH_4},\,\Lambda_u)} \\ I_{(\Lambda_1,\,\Lambda_1)} & \mathbf{0}_{(\Lambda_1,\,\Lambda_2)} & \mathbf{0}_{(\Lambda_1,\,\Lambda_3^+)} & \mathbf{0}_{(\Lambda_1,\,\Lambda_3^c)} & \mathbf{0}_{(\Lambda_1,\,\Lambda_u)} \\ \mathbf{0}_{(\Lambda_2,\,\Lambda_1)} & I_{(\Lambda_2,\,\Lambda_2)} & \mathbf{0}_{(\Lambda_2,\,\Lambda_3^+)} & \mathbf{0}_{(\Lambda_2,\,\Lambda_3^c)} & \mathbf{0}_{(\Lambda_2,\,\Lambda_u)} \\ \mathbf{0}_{(\Lambda_3^+,\,\Lambda_1)} & \mathbf{0}_{(\Lambda_3^+,\,\Lambda_2)} & I_{(\Lambda_3^+,\,\Lambda_3^+)} & \mathbf{0}_{(\Lambda_3^+,\,\Lambda_3^c)} & \mathbf{0}_{(\Lambda_3^+,\,\Lambda_u)} \end{bmatrix},$$

and assume that we can find some vectors $\overline{\rho}^g \in \mathbb{R}^{S_3}$ and $\overline{\rho}^g \geq \mathbf{0}$, $\overline{\rho}^h \in \mathbb{R}^{W_2}$ and $\overline{\rho}^h \geq \mathbf{0}$, $\overline{\rho}^i \in \mathbb{R}^{W_2}$ and $\overline{\rho}^i \geq \mathbf{0}$, $\overline{\rho}^j \in \mathbb{R}^{U_3}$ and $\overline{\rho}^j \geq \mathbf{0}$, $\overline{\rho}^m \in \mathbb{R}^{W_3}$ and $\overline{\rho}^m \geq \mathbf{0}$, and $\overline{\rho}^n \in \mathbb{R}^{U_4}$ and $\overline{\rho}^n \geq \mathbf{0}$, such that

$$\sum_{s=1}^{S_3} \rho_s^g \begin{bmatrix} \begin{pmatrix} BB^\top \end{pmatrix}^\top_{(g_s,\,\cdot)} \\ \begin{bmatrix} \mathbf{0}_{D_1+D_2} \\ e^{S_3}_{g_s} \end{bmatrix} \end{bmatrix} + \sum_{s=1}^{W_2} \rho_s^h \begin{bmatrix} \begin{pmatrix} BB^\top \end{pmatrix}^\top_{(h_s,\,\cdot)} \\ \begin{bmatrix} e^{W_2}_{h_s} \\ \mathbf{0}_{Tm_2-D_1} \end{bmatrix} \end{bmatrix} + \sum_{s=1}^{W_2} \rho_s^i \begin{bmatrix} \begin{bmatrix} e^{W_2}_{i_s} \\ \mathbf{0}_{Tm_2-D_1} \end{bmatrix} \\ \mathbf{0}_{Tm_2} \end{bmatrix} +$$
$$\sum_{s=1}^{U_3} \rho_s^j \begin{bmatrix} \begin{bmatrix} \mathbf{0}_{D_1} \\ e^{U_3}_{j_s} \\ \mathbf{0}_{D_3+D_4+D_5} \end{bmatrix} \\ \mathbf{0}_{Tm_2} \end{bmatrix} + \sum_{s=1}^{W_3} \rho_s^m \begin{bmatrix} \mathbf{0}_{Tm_2} \\ \begin{bmatrix} \mathbf{0}_{(D_1+D_2+D_3)} \\ e^{W_3}_{m_s} \\ \mathbf{0}_{D_5} \end{bmatrix} \end{bmatrix} + \sum_{s=1}^{U_4} \rho_s^n \begin{bmatrix} \mathbf{0}_{Tm_2} \\ \begin{bmatrix} e^{U_4}_{n_s} \\ \mathbf{0}_{D_4+D_5} \end{bmatrix} \end{bmatrix} = \mathbf{0}.$$

The above equation is equivalent to the compact system

$$\begin{bmatrix} \sum_{s=1}^{S_3} \rho_s^g \left(\left(BB^\top \right)_{(g_s,\,\cdot)} \right)^\top + \sum_{s=1}^{W_2} \rho_s^h \left(\left(BB^\top \right)_{(h_s,\,\cdot)} \right)^\top + \begin{bmatrix} \overline{\rho}^i \\ \overline{\rho}^j \\ \mathbf{0}_{D_3+D_4+D_5} \end{bmatrix} \\ \overline{\rho}^h + \overline{\rho}^n_{\Lambda_1} \\ \overline{\rho}^n_{\Lambda_2} \\ \overline{\rho}^g_{\Lambda_3^+} + \overline{\rho}^n_{\Lambda_3^+} \\ \overline{\rho}^g_{\Lambda_3^c} + \overline{\rho}^m \\ \overline{\rho}^g_{\Lambda_u} \end{bmatrix} = \mathbf{0},$$

which leads to $\overline{\rho}^g = \mathbf{0}$, $\overline{\rho}^h = \mathbf{0}$, $\overline{\rho}^i = \mathbf{0}$, $\overline{\rho}^j = \mathbf{0}$, $\overline{\rho}^m = \mathbf{0}$, $\overline{\rho}^n = \mathbf{0}$, given that $\overline{\rho}^g \geq \mathbf{0}$, $\overline{\rho}^h \geq \mathbf{0}$, $\overline{\rho}^i \geq \mathbf{0}$, $\overline{\rho}^j \geq \mathbf{0}$, $\overline{\rho}^m \geq \mathbf{0}$, $\overline{\rho}^n \geq \mathbf{0}$. Therefore, the row vectors in the matrix Γ_{sub} are positively-linearly independent. □

4.4 The main result

Based on the above lemmas, we are ready to present the main theorem on the MPEC–MFCQ.

Theorem 3 *Let $v = (C, \zeta, z, \alpha, \xi)$ be any feasible point for the MPEC (26), then v satisfies the MPEC–MFCQ.*

Proof Assume there exist $\overline{\rho}^a \in \mathbb{R}^{S_1}$ and $\overline{\rho}^a \geq \mathbf{0}$, $\overline{\rho}^b \in \mathbb{R}^{W_1}$ and $\overline{\rho}^b \geq \mathbf{0}$, $\overline{\rho}^c \in \mathbb{R}^{W_1}$ and $\overline{\rho}^c \geq \mathbf{0}$, $\overline{\rho}^d \in \mathbb{R}^{U_1}$ and $\overline{\rho}^d \geq \mathbf{0}$, $\overline{\rho}^e \in \mathbb{R}^{S_2}$ and $\overline{\rho}^e \geq \mathbf{0}$, $\overline{\rho}^f \in \mathbb{R}^{U_2}$ and $\overline{\rho}^f \geq \mathbf{0}$, $\overline{\rho}^g \in \mathbb{R}^{S_3}$ and $\overline{\rho}^g \geq \mathbf{0}$, $\overline{\rho}^h \in \mathbb{R}^{W_2}$ and $\overline{\rho}^h \geq \mathbf{0}$, $\overline{\rho}^i \in \mathbb{R}^{W_2}$ and $\overline{\rho}^i \geq \mathbf{0}$, $\overline{\rho}^j \in \mathbb{R}^{U_3}$ and $\overline{\rho}^j \geq \mathbf{0}$, $\overline{\rho}^k \in \mathbb{R}^{S_4}$ and $\overline{\rho}^k \geq \mathbf{0}$, $\overline{\rho}^l \in \mathbb{R}^{W_3}$ and $\overline{\rho}^l \geq \mathbf{0}$, $\overline{\rho}^m \in \mathbb{R}^{W_3}$ and $\overline{\rho}^m \geq \mathbf{0}$, $\overline{\rho}^n \in \mathbb{R}^{U_4}$ and $\overline{\rho}^n \geq \mathbf{0}$, such that the following holds

$$\sum_{s=1}^{S_1} \rho_s^a \begin{bmatrix} 0 \\ \mathbf{0}_{Tm_1} \\ \left(\Gamma_a^3\right)^\top_{(a_s,\,\cdot)} \\ \left(AB^\top\right)^\top_{(a_s,\,\cdot)} \\ \mathbf{0}_{Tm_2} \end{bmatrix} + \sum_{s=1}^{W_1} \rho_s^b \begin{bmatrix} 0 \\ \mathbf{0}_{Tm_1} \\ \left(\Gamma_b^3\right)^\top_{(b_s,\,\cdot)} \\ \left(AB^\top\right)^\top_{(b_s,\,\cdot)} \\ \mathbf{0}_{Tm_2} \end{bmatrix} + \sum_{s=1}^{W_1} \rho_s^c \begin{bmatrix} 0 \\ \left(\Gamma_c^2\right)^\top_{(c_s,\,\cdot)} \\ \mathbf{0}_{Tm_1} \\ \mathbf{0}_{Tm_2} \\ \mathbf{0}_{Tm_2} \end{bmatrix}$$

$$+ \sum_{s=1}^{U_1} \rho_s^d \begin{bmatrix} 0 \\ \left(\Gamma_d^2\right)^\top_{(d_s,\,\cdot)} \\ \mathbf{0}_{Tm_1} \\ \mathbf{0}_{Tm_2} \\ \mathbf{0}_{Tm_2} \end{bmatrix} + \sum_{s=1}^{S_2} \rho_s^e \begin{bmatrix} 0 \\ \left(\Gamma_e^2\right)^\top_{(e_s,\,\cdot)} \\ \mathbf{0}_{Tm_1} \\ \mathbf{0}_{Tm_2} \\ \mathbf{0}_{Tm_2} \end{bmatrix} + \sum_{s=1}^{U_2} \rho_s^f \begin{bmatrix} 0 \\ \mathbf{0}_{Tm_1} \\ \left(\Gamma_f^3\right)^\top_{(f_s,\,\cdot)} \\ \mathbf{0}_{Tm_2} \\ \mathbf{0}_{Tm_2} \end{bmatrix}$$

$$
+\sum_{s=1}^{S_3}\rho_s^g\begin{bmatrix}0\\ \mathbf{0}_{Tm_1}\\ \mathbf{0}_{Tm_1}\\ \left(BB^\top\right)^\top_{(g_s,\,\cdot\,)}\\ \left(\Gamma^5_g\right)^\top_{(g_s,\,\cdot\,)}\end{bmatrix}+\sum_{s=1}^{W_2}\rho_s^h\begin{bmatrix}0\\ \mathbf{0}_{Tm_1}\\ \mathbf{0}_{Tm_1}\\ \left(BB^\top\right)^\top_{(h_s,\,\cdot\,)}\\ \left(\Gamma^5_h\right)^\top_{(h_s,\,\cdot\,)}\end{bmatrix}+\sum_{s=1}^{W_2}\rho_s^i\begin{bmatrix}0\\ \mathbf{0}_{Tm_1}\\ \mathbf{0}_{Tm_1}\\ \left(\Gamma^4_i\right)^\top_{(i_s,\,\cdot\,)}\\ \mathbf{0}_{Tm_2}\end{bmatrix}
$$

$$
+\sum_{s=1}^{U_3}\rho_s^j\begin{bmatrix}0\\ \mathbf{0}_{Tm_1}\\ \mathbf{0}_{Tm_1}\\ \left(\Gamma^4_j\right)^\top_{(j_s,\,\cdot\,)}\\ \mathbf{0}_{Tm_2}\end{bmatrix}+\sum_{s=1}^{S_4}\rho_s^k\begin{bmatrix}1\\ \mathbf{0}_{Tm_1}\\ \mathbf{0}_{Tm_1}\\ \left(\Gamma^4_k\right)^\top_{(k_s,\,\cdot\,)}\\ \mathbf{0}_{Tm_2}\end{bmatrix}+\sum_{s=1}^{W_3}\rho_s^l\begin{bmatrix}1\\ \mathbf{0}_{Tm_1}\\ \mathbf{0}_{Tm_1}\\ \left(\Gamma^4_l\right)^\top_{(l_s,\,\cdot\,)}\\ \mathbf{0}_{Tm_2}\end{bmatrix}
$$

$$
+\sum_{s=1}^{W_3}\rho_s^m\begin{bmatrix}0\\ \mathbf{0}_{Tm_1}\\ \mathbf{0}_{Tm_1}\\ \mathbf{0}_{Tm_2}\\ \left(\Gamma^5_m\right)^\top_{(m_s,\,\cdot\,)}\end{bmatrix}+\sum_{s=1}^{U_4}\rho_s^n\begin{bmatrix}0\\ \mathbf{0}_{Tm_1}\\ \mathbf{0}_{Tm_1}\\ \mathbf{0}_{Tm_2}\\ \left(\Gamma^5_n\right)^\top_{(n_s,\,\cdot\,)}\end{bmatrix}=\mathbf{0}. \tag{43}
$$

From the first row in Eq. (43), we get $\sum_{s=1}^{S_4}\rho_s^k+\sum_{s=1}^{W_3}\rho_s^l=0$. Together with the fact that $\overline{\rho}^k\geq\mathbf{0}$, $\overline{\rho}^l\geq\mathbf{0}$, we get $\overline{\rho}^k=\mathbf{0}$ and $\overline{\rho}^l=\mathbf{0}$. From Lemma 1, we get $\overline{\rho}^c=\mathbf{0}$, $\overline{\rho}^d=\mathbf{0}$, $\overline{\rho}^e=\mathbf{0}$ in Equation (43). From Lemma 2, we get $\overline{\rho}^a=\mathbf{0}$, $\overline{\rho}^b=\mathbf{0}$, $\overline{\rho}^f=\mathbf{0}$ in Eq. (43). From Lemma 3, we get $\overline{\rho}^g=\mathbf{0}$, $\overline{\rho}^h=\mathbf{0}$, $\overline{\rho}^i=\mathbf{0}$, $\overline{\rho}^j=\mathbf{0}$, $\overline{\rho}^m=\mathbf{0}$, $\overline{\rho}^n=\mathbf{0}$ in Eq. (43).

In summary, the row vectors in the matrix Γ (35) are positive-linearly independent at every feasible point v for the MPEC (26). That is to say, every feasible point v for the MPEC (26) satisfies the MPEC-MFCQ. □

5 Numerical results

In this section, we present the GR–CV, which is a concrete implementation of the GRM in Algorithm 1 for selecting the hyperparameter C in SVC, as shown in Algorithm 2. We show numerical results of the proposed GR–CV, and compare it with other approaches.

Algorithm 2 The Global Relaxation Cross-Validation Algorithm (GR-CV)

1: Given T, split the data set into a subset Ω with l_1 points and a hold-out test set Θ with l_2 points. The set Ω is equally partitioned into T pairwise disjoint subsets, one for each fold.
2: **Select** an optimal hyperparameter $\widehat{C}$ by the GRM in Algorithm 1.
3: **Post-processing procedure.** The regularization hyperparameter $\widehat{C}$ is rescaled by a factor $\frac{T}{T-1}$. Then, an l_1-loss SVC problem is solved on the subset Ω using $\frac{T}{T-1}\widehat{C}$ by ALM-SNCG algorithm in Yan and Li (2020). This gives the final classifier $\widehat{w}$.

Table 1 Descriptions of data sets

Data set	l_1	l_2	n	Data set	l_1	l_2	n
Heart	189	81	13	splice	300	700	60
Breast-cancer	240	172	10	fourclass	300	562	2
Colon-cancer	36	26	2000	w1a	240	260	300
Ionosphere	246	105	34	w2a	300	500	300
Australian	270	420	14	a1a	300	200	119
Diabetes	270	498	8	german.number	207	793	24

All the numerical tests are conducted in Matlab R2018a on a Windows 7 Dell Laptop with an Intel(R) Core(TM) i5-6500U CPU at 3.20GHz and 8 GB of RAM. All the data sets are collected from the LIBSVM library: https://www.csie.ntu.edu.tw/cjlin/libsvmtools/datasets/. Each data set is split into a subset Ω with l_1 points (it is used for cross-validation) and a hold-out test set Θ with l_2 points. The data descriptions are shown in Table 1.

We compare our GR–CV with two other approaches: the inexact cross-validation method (In-CV) and the grid search method (G–S). In-CV Kunapuli et al. (2008b) is a relaxation method based on the relaxation of the complementarity constraints by a prescribed tolerance parameter **tol** > 0. That is, solving (NLP-t_k) with $t_k =$ **tol** as a fixed tolerance rather than decreasing t_k gradually.

The parameters of three methods are set as follows. For GR–CV, we set the initial values as $v_0 = \left[1,\ \mathbf{0}_{1\times\overline{m}}\right]^\top$, $t_0 = 1$, $t_{\min} = 10^{-8}$, $\sigma = 0.01$. The relaxed subproblems (NLP-t_k) are solved by the `snsolve` function, which is part of the SNOPT solver (Gill et al. 2002). For In-CV, we use the same v_0 as in GR–CV and **tol** $= 10^{-4}$. For G–S, we use $C \in \{10^{-4},\ 10^{-3},\ 10^{-2}, 10^{-1},\ 1,\ 10^{1},\ 10^{2},\ 10^{3},\ 10^{4}\}$, which is a commonly used grid range (Bennett et al. 2006; Kunapuli et al. 2008b; Kunapuli 2008; Moore et al. 2009). In each training process, the ALM–SNCG algorithm from Yan and Li (2020), which is outstanding and competitive with the most popular methods in LIBLINEAR (https://www.csie.ntu.edu.tw/~cjlin/liblinear/) in both speed and accuracy, is used to solve the l_1-loss SVC problem.

We compare the aforementioned methods in the following three aspects:

1. Test error (E_t) as defined by

$$E_t = \frac{1}{l_2} \sum_{(x,y)\in\Theta} \frac{1}{2} \mid \text{sign}\left(\widehat{w}^\top x\right) - y \mid,$$

   which is a measure of the ability of generalization.
2. CV error (E_C) as defined in the objective function of problem (14).
3. The number of iterations k for an algorithm, and the total number of iterations it for solving the subproblems (short for $(k,\ it)$).

We also report the maximum violation of all constraints defined as in (27), to measure the feasibility of the final solution given by GR–CV and In-CV.

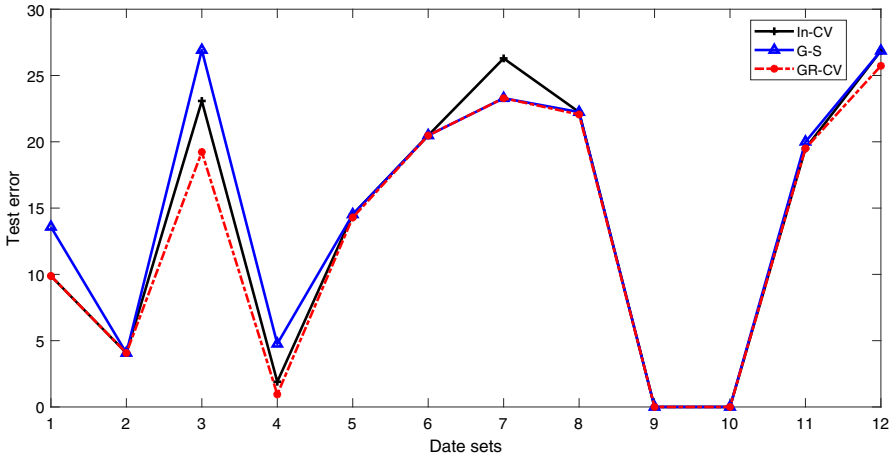


Fig. 9 The comparison among the three methods on test error

The results are reported in Table 2, where we mark the winners of test error E_t, CV error E_C and the maximum violation of all constraints Vio in bold. We also show the comparisons of the three methods for different data sets on test error E_t and CV error E_C in Figs. 9 and 10, respectively. The data sets on the horizontal axis are arranged in the order shown in Table 2.

From Figs. 9, 10 and Table 2, we have the following observations. Firstly, GR–CV performs the best in terms of test error in Fig. 9, implying that our approach is more capable of generalization. Secondly, in terms of CV error in Fig. 10, GR–CV is competitive with G–S. GR–CV is the winner in five data sets of all the twelve datasets whereas G-S wins in eight datasets among the twelve datasets. Finally, comparing GR–CV with In-CV, the feasibility of the solution returned by GR–CV is significantly better than that by In-CV since Vio given by GR–CV is much smaller than that by In-CV. In terms of cpu time, it is obvious that In-CV takes less time than GR-CV since it only solves the relaxation problem (NLP-t_k) once. Since G-S is basically solving a completely different type of problem to find the hyperparameter C, it doesn't make sense to compare the cpu time between GR–CV and G–S.

To further study the effect of increasing the number of folds on test error E_t and CV error E_C in the three methods, we report the results on the Australian data set in Fig. 11. The results show that as T changes, the test error for GR–CV is always the lowest, and the CV error for GR–CV is competitive with the other two methods. Meanwhile it is clear that larger number of folds can be successfully solved for GR–CV, the computing time grows with the number of folds because of the increasing number of variables and constraints for the MPEC to be solved. The ranges of the test error and CV error for different numbers of folds are not large, so $T = 3$ represents a reasonable choice.

Table 2 Computational results for $T = 3$

	Data set	Method	E_t (%)	E_C (%)	Vio	$(k,\ it)$
1	Heart	GR-CV	**9.88**	**17.46**	**1.51e−6**	(5, 24165)
		In-CV	**9.88**	17.95	0.010	(1,12418)
		G-S	13.58	**17.46**	–	(27,425)
2	Breast-cancer	GR-CV	**4.07**	**5.42**	**4.98e−4**	(5,17092)
		In-CV	**4.07**	**5.42**	0.006	(1,14971)
		G-S	**4.07**	6.25	–	(27,298)
3	Colon-cancer	GR-CV	**19.23**	**2.78**	**9.69e−5**	(5,2166)
		In-CV	23.08	**2.78**	0.005	(1,1102)
		G-S	26.92	**2.78**	–	(27,167)
4	Ionosphere	GR-CV	**0.95**	27.61	**0.03**	(5,96200)
		In-CV	1.90	29.76	**0.03**	(1,29530)
		G-S	4.76	**18.70**	–	(27,522)
5	Australian	GR-CV	**14.29**	**14.44**	**3.03e−6**	(5,32583)
		In-CV	14.52	14.81	0.008	(1,26703)
		G-S	14.52	**14.44**	–	(27,430)
6	Diabetes	GR-CV	**20.48**	**24.44**	**1.75e−5**	(5,33294)
		In-CV	**20.48**	25.18	0.005	(1,26558)
		G-S	**20.48**	25.19	–	(27,416)
7	Splice	GR-CV	**23.29**	29.01	0.009	(5,83306)
		In-CV	26.29	24.63	**0.005**	(1,24333)
		G-S	**23.29**	**23.33**	–	(27,526)
8	Fourclass	GR-CV	**22.06**	28.67	**5.83e−5**	(5,17275)
		In-CV	22.24	28.65	0.008	(1,8989)
		G-S	22.24	**23.33**	–	(27,349)
9	W1a	GR-CV	**0.00**	23.33	**4.26e−4**	(5,75793)
		In-CV	**0.00**	**22.88**	0.009	(1,28810)
		G-S	**0.00**	30.00	–	(27,366)
10	W2a	GR-CV	**0.00**	25.93	**1.50e−4**	(5,88758)
		In-CV	**0.00**	**22.11**	0.009	(1,31708)
		G-S	**0.00**	35.67	–	(27,522)
11	A1a	GR-CV	**19.50**	15.33	**7.64e−5**	(5,64349)
		In-CV	**19.50**	15.65	0.013	(1,36010)
		G-S	20.00	**14.67**	–	(27,533)
12	german.	GR-CV	**25.73**	26.09	**5.29e−5**	(5,33317)
	Number	In-CV	26.86	26.08	0.068	(1,24850)
		G-S	26.86	**25.60**	–	(27,482)

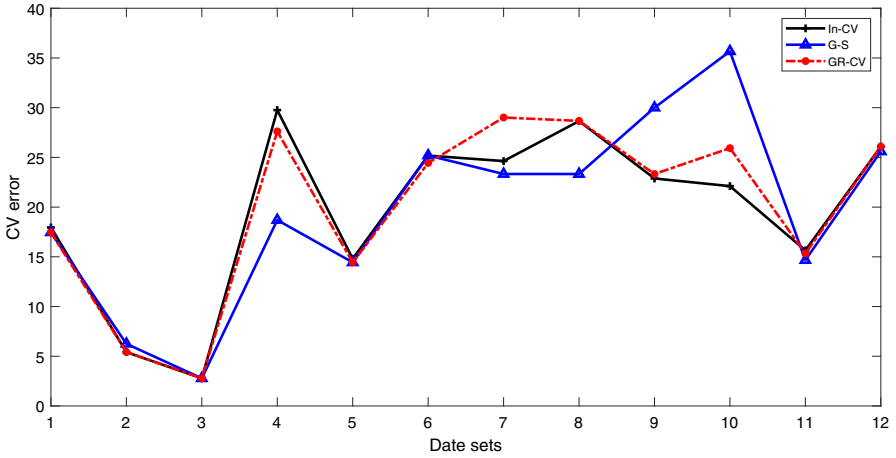


Fig. 10 The comparison among the three methods on CV error

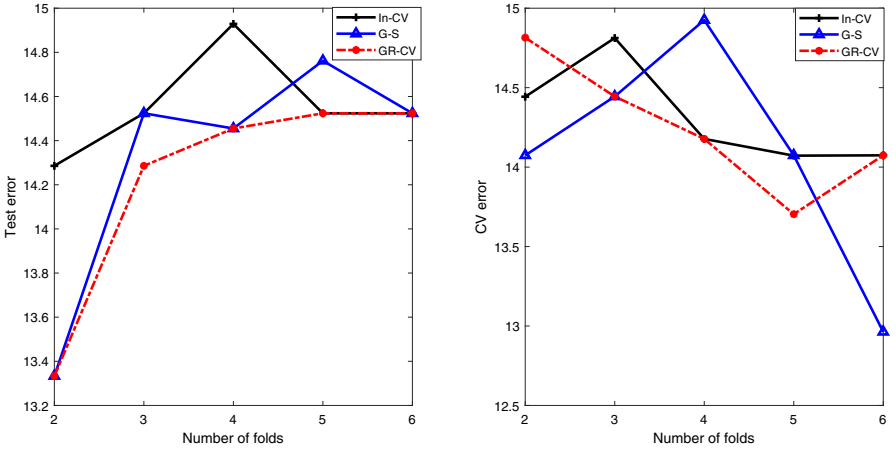


Fig. 11 Effect of increasing the number of folds on test error and CV error

6 Conclusion

In this paper, we have proposed a bilevel optimization model for the hyperparameter selection for support vector classification in which the upper-level problem minimizes a T-fold cross-validation error and the lower-level problems are T l_1-loss SVC problems on the training sets. We reformulated the bilevel optimization problem into an MPEC, and proposed the GR–CV to solve it based on the GRM from Scholtes (2001). We also proved that the MPEC–MFCQ automatically holds at each feasible point. Extensive numerical results on the data sets from the LIBSVM library demonstrated the superior generalization performance of the proposed method over almost all the data sets used in this paper. The proposed approach has the potential to deal with other hyperparameter selection problems in SVM, which may involve multiple hyper-

parameters or other types of loss functions. However, whether the resulting MPEC enjoys the property of MPEC–MFCQ needs to be further investigated. How to choose the most suitable numerical algorithms to solve the perturbed problem resulting from the Scholtes relaxation is also worth further study. These topics will be investigated further in the near future.

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

References

Anitescu M (2000) On solving mathematical programs with complementarity constraints as nonlinear programs. Preprint ANL/MCS-P864-1200, Argonne National Laboratory, Argonne, IL **3**

Bennett KP, Hu J, Ji XY, Kunapuli G, Pang J-S (2006) Model selection via bilevel optimization. In: The 2006 IEEE International Joint Conference on Neural Network Proceedings, pp 1922–1929 . IEEE

Bennett KP, Kunapuli G, Hu J, Pang J-S (2008) Bilevel optimization and machine learning. In: IEEE World Congress on Computational Intelligence, pp 25–47

Chapelle O, Vapnik V, Bousquet O, Mukherjee S (2002) Choosing multiple parameters for support vector machines. Mach Learn 46(1):131–159

Chauhan VK, Dahiya K, Sharma A (2019) Problem formulations and solvers in linear SVM: a review. Artif Intell Rev 52(2):803–855

Colson B, Marcotte P, Savard G (2007) An overview of bilevel optimization. Ann Oper Res 153(1):235–256

Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297

Couellan N, Wang WJ (2015) Bi-level stochastic gradient for large scale support vector machine. Neurocomputing 153:300–308

Cristianini N, Shawe-Taylor J (2000) An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press, Cambridge

Crockett C, Fessler JA (2021) Bilevel methods for image reconstruction. arXiv preprint arXiv:2109.09610

Dempe S (2002) Foundations of Bilevel Programming. Springer, New York

Dempe S (2003) Annotated bibliography on bilevel programming and mathematical programs with equilibrium constraints. Optimization 52(3):333–359

Dempe S, Zemkoho AB (2012) On the Karush-Kuhn-Tucker reformulation of the bilevel optimization problem. Nonlinear Analysis: Theory, Methods Appl 75(3):1202–1218

Dempe S, Zemkoho AB (2020) Bilevel Optimization Advances and Next Challenges. Springer, New York

Dong Y-L, Xia Z-Q, Wang M-Z (2007) An MPEC model for selecting optimal parameter in support vector machines. In: The First International Symposium on Optimization and Systems Biology, pp 351–357

Duan KB, Keerthi SS, Poo AN (2003) Evaluation of simple performance measures for tuning SVM hyperparameters. Neurocomputing 51:41–59

Facchinei F, Pang J-S (2007) Finite-dimensional Variational Inequalities and Complementarity Problems. Springer, NewYork

Fischer A, Zemkoho AB, Zhou S (2021) Semismooth Newton-type method for bilevel optimization: global convergence and extensive numerical experiments. Optimization Methods & Software, 1–35

Flegel ML (2005) Constraint qualifications and stationarity concepts for mathematical programs with equilibrium constraints. PhD thesis, Universität Würzburg

Fletcher R, Leyffer S, Ralph D, Scholtes S (2006) Local convergence of SQP methods for mathematical programs with equilibrium constraints. SIAM J Optim 17(1):259–286
Fukushima M, Tseng P (2002) An implementable active-set algorithm for computing a B-stationary point of a mathematical program with linear complementarity constraints. SIAM J Optim 12(3):724–739
Galli L, Lin C-J (2021) A study on truncated newton methods for linear classification. IEEE Transactions on Neural Networks and Learning Systems
Gill PE, Murray W, Saunders MA (2002) User's Guide for Snopt version 6. A Fortran Package for Large-Scale Nonlinear Programming. University of California, California
Guo L, Lin G-H, Ye JJ (2015) Solving mathematical programs with equilibrium constraints. J Optim Theory Appl 166(1):234–256
Harder F, Mehlitz P, Wachsmuth G (2021) Reformulation of the M-stationarity conditions as a system of discontinuous equations and its solution by a semismooth Newton method. SIAM J Optim 31(2):1459–1488
Hoheisel T, Kanzow C, Schwartz A (2013) Theoretical and numerical comparison of relaxation methods for mathematical programs with complementarity constraints. Math Program 137(1):257–288
Hsieh C-J, Chang K-W, Lin C-J, Keerthi SS, Sundararajan S (2008) A dual coordinate descent method for large-scale linear SVM. In Proceedings of the 25th International Conference on Machine Learning, pp 408–415
Huang X, Shi L, Suykens JA (2013) Support vector machine classifier with pinball loss. IEEE Trans Pattern Anal Mach Intell 36(5):984–997
Jara-Moroni F, Pang J-S, Wächter A (2018) A study of the difference-of-convex approach for solving linear programs with complementarity constraints. Math Program 169(1):221–254
Júdice JJ (2012) Algorithms for linear programming with linear complementarity constraints. TOP 20(1):4–25
Keerthi S, Sindhwani V, Chapelle O (2006) An efficient method for gradient-based adaptation of hyperparameters in SVM models. Advances in neural information processing systems **19**
Kunapuli G (2008) A Bilevel Optimization Approach to Machine Learning. Rensselaer Polytechnic Institute, New York
Kunapuli G, Bennett KP, Hu J, Pang J-S (2008) Classification model selection via bilevel programming. Optim Methods Softw 23(4):475–489
Kunapuli G, Bennett KP, Hu J, Pang J-S (2008) Bilevel model selection for support vector machines. Data mining mathe programming 45:129–158
Kunisch K, Pock T (2013) A bilevel optimization approach for parameter learning in variational models. SIAM J Imag Sci 6(2):938–983
Lee Y-C, Pang J-S, Mitchell JE (2015) Global resolution of the support vector machine regression parameters selection problem with LPCC. EURO J Comput Optim 3(3):197–261
Li JL, Huang RS, Jian JB (2015) A superlinearly convergent QP-free algorithm for mathematical programs with equilibrium constraints. Appl Math Comput 269:885–903
Lin G-H, Xu MW, Ye JJ (2014) On solving simple bilevel programs with a nonconvex lower level program. Math Program 144(1):277–305
Luo G (2016) A review of automatic selection methods for machine learning algorithms and hyper-parameter values. Netw Model Anal Health Inform Bioinform 5(1):1–16
Luo Z-Q, Pang J-S, Ralph D (1996) Mathematical Programs with Equilibrium Constraints. Cambridge University Press, Cambridge
Mangasarian OL (1994) Misclassification minimization. J Global Optim 5(4):309–323
Mejía-de-Dios J-A, Mezura-Montes E (2019) A metaheuristic for bilevel optimization using tykhonov regularization and the quasi-newton method. In: 2019 IEEE Congress on Evolutionary Computation (CEC), pp 3134–3141
Momma M, Bennett KP (2002) A pattern search method for model selection of support vector regression. In: Proceedings of the 2002 SIAM International Conference on Data Mining, pp 261–274
Moore G, Bergeron C, Bennett KP. Gradient-type methods for primal SVM model selection
Moore G, Bergeron C, Bennett KP (2009) Nonsmooth bilevel programming for hyperparameter selection. In: 2009 IEEE International Conference on Data Mining Workshops, pp 374–381
Ochs P, Ranftl R, Brox T, Pock T (2016) Techniques for gradient-based bilevel optimization with non-smooth lower level problems. J Mathe Imag Vision 56(2):175–194
Ochs P, Ranftl R, Brox T, Pock T (2015) Bilevel optimization with nonsmooth lower level problems. In: International Conference on Scale Space and Variational Methods in Computer Vision, pp 654–665

Okuno T, Takeda A, Kawana A (2018) Hyperparameter learning for bilevel nonsmooth optimization. arXiv preprint arXiv:1806.01520
Scholtes S (2001) Convergence properties of a regularization scheme for mathematical programs with complementarity constraints. SIAM J Optim 11(4):918–936
Shalev-Shwartz S, Singer Y, Srebro N, Cotter A (2011) Pegasos: Primal estimated sub-gradient solver for SVM. Math Program 127(1):3–30
Vapnik V (2013) The Nature of Statistical Learning Theory. Springer, New York
Wang H, Shao Y, Zhou S, Zhang C, Xiu N (2021) Support vector machine classifier via $L_{0/1}$ soft-margin loss. IEEE Transactions on Pattern Analysis and Machine Intelligence
Wu J, Zhang LW, Zhang Y (2015) An inexact Newton method for stationary points of mathematical programs constrained by parameterized quasi-variational inequalities. Numer Algorithms 69(4):713–735
Yan YQ, Li QN (2020) An efficient augmented Lagrangian method for support vector machine. Optim Methods Softw 35(4):855–883
Ye JJ (2005) Necessary and sufficient optimality conditions for mathematical programs with equilibrium constraints. J Math Anal Appl 307(1):350–369
Ye JJ, Zhu DL (2010) New necessary optimality conditions for bilevel programs by combining the MPEC and value function approaches. SIAM J Optim 20(4):1885–1905
Yu B, Mitchell JE, Pang J-S (2019) Solving linear programs with complementarity constraints using branch-and-cut. Math Program Comput 11(2):267–310
Yu T, Zhu H (2020) Hyper-parameter optimization: A review of algorithms and applications. arXiv preprint arXiv:2003.05689
Zemkoho AB, Zhou SL (2021) Theoretical and numerical comparison of the karush-kuhn-tucker and value function reformulations in bilevel optimization. Comput Optim Appl 78(2):625–674
Zhang T (2004) Solving large scale linear prediction problems using stochastic gradient descent algorithms. In: Proceedings of the Twenty-first International Conference on Machine Learning, p 116