Two-way Cutting-plane Algorithm for Best Subset Selection Considering Multicollinearity

Anonymous authors
Paper under double-blind review

Abstract

When linear dependence exists between some explanatory variables in a regression model, the estimates of regression coefficients become unstable, thereby making the interpretation of the estimation results unreliable. To eliminate such multicollinearity, we propose a high-performance method for selecting the best subset of explanatory variables for linear and logistic regression models. Specifically, we first derive a bilevel reformulation of the optimization problem for best subset selection with a multicollinearity constraint. We then develop a two-way cutting-plane algorithm that uses cutting planes in two ways; one type of cutting planes is used to approximate an upper-level nonlinear objective function, and the other type of cutting planes is used to remove subsets with collinearity. We prove that this algorithm outputs a solution with guaranteed global optimality within a finite number of iterations. Computational results based on synthetic and public datasets demonstrate the effectiveness of our method by comparison with the L1-regularized estimation and the previous cutting-plane algorithm for eliminating multicollinearity. Our method is not only a fast computational framework for best subset selection, but also has the advantage of improving the reliability of regression analysis by eliminating multicollinearity.

1 Introduction

1.1 Background

In linear and logistic regression models, the response variable is predicted using a linear combination of the explanatory variables. Specifically, linear regression is applicable when the response variable is real-valued (normally distributed), whereas logistic regression is appropriate when the response variable is binary-valued (Bernoulli distributed). When there are many irrelevant variables, however, subset selection improves the interpretability of regression models by selecting a relevant subset of explanatory variables (James et al., 2014). Subset selection has many other benefits, such as reducing overfitting and improving predictive performance (Guyon & Elisseeff, 2003). For these reasons, subset selection has long been considered as an important problem in statistics (Hocking, 1976), and it has also attracted considerable attention in machine learning with the primary aim of improving predictive performance (Guyon & Elisseeff, 2003).

When linear dependence exists between some of the explanatory variables, the estimates of regression coefficients become unstable, thereby making the interpretation of the estimation results unreliable. This problem is known as multicollinearity (Chatterjee & Hadi, 2015; James et al., 2014). There are several approaches to avoiding the adverse effects of multicollinearity (Chatterjee & Hadi, 2015). One approach is the orthogonal transformation, used in principal component regression (Jolliffe, 1982) and partial least squares regression (Wold et al., 2001). This approach can increase the computational stability of model estimation, but at the expense of worse predictive performance and greater susceptibility to outliers. Another approach is the regularized estimation, such as ridge regression (Hoerl & Kennard, 1970). This approach also helps prevent regression models from overfitting to noisy datasets; however, the regularization terms produce biased estimates, which are undesirable from the model interpretation point of view. For these reasons, subset selection, which maintains high predictive performance and prevents biased estimates, is also the most commonly used method to avoid multicollinearity.

While various methods have been proposed for subset selection (Chandrashekar & Sahin, 2014; Guyon & Elisseeff, 2003; Li et al., 2017; Liu & Motoda, 2007), we focus particularly on the mixed-integer optimization (MIO) approach to best subset selection (Bertsimas et al., 2016; Cozad et al., 2014; Hastie et al., 2020; Konno & Yamamoto, 2009; Ustun & Rudin, 2016). In recent years, advances in optimization algorithms and computer hardware have established MIO techniques as a practical solution methodology (Bixby, 2012; Koch et al., 2022). Bertsimas & King (2016) states that the mindset that MIO is theoretically elegant but practically irrelevant is no longer supported. In fact, applications of MIO techniques are progressing rapidly in a variety of statistical and machine learning tasks (Carrizosa et al., 2021; Gambella et al., 2021). Compared with many heuristic algorithms, the MIO approach has the advantage of selecting the best subset of explanatory variables with respect to given criterion functions (Miyashiro & Takano, 2015b;a; Park & Klabjan, 2020; Takano & Miyashiro, 2020). MIO methods for subset selection have been extended to logistic regression (Ahari & Kocuk, 2023; Bertsimas & King, 2017; Sato et al., 2016), ordinal regression (Naganuma et al., 2019; Sato et al., 2017), count regression (Saishu et al., 2021), survival analysis (Saishu et al., 2024), support vector machines (Maldonado et al., 2014; Tamura et al., 2024), and dimensionality reduction (Berk & Bertsimas, 2019; Watanabe et al., 2023).

1.2 Related work

Prior studies have proposed various exact and heuristic algorithms using MIO techniques for subset selection. Bertsimas et al. (2020; 2021) developed high-performance cutting-plane algorithms based on bilevel reformulation for best subset selection in linear and logistic regression models. Hazimeh & Mazumder (2020) designed a fast subset selection algorithm based on coordinate descent and local combinatorial optimization for linear regression. Xie & Deng (2020) designed greedy and randomized algorithms based on mixed-integer second-order cone optimization formulations for sparse ridge regression. Hazimeh et al. (2022) established a specialized branch-and-bound framework for best subset selection by applying an efficient coordinate descent to node relaxations for linear regression. Liu et al. (2024) built a customized branch-and-bound framework using an efficient lower bound calculation and heuristic search for sparse ridge regression. Dedieu et al. (2021) devised two types of algorithms for subset selection in classification models: an exact algorithm based on integrality generation and a heuristic algorithm based on coordinate descent and local combinatorial search. However, these high-performance algorithms do not directly account for multicollinearity and therefore cannot guarantee that the selected subset does not contain collinear variables.

Various MIO methods have been proposed for subset selection to eliminate multicollinearity from machine learning models (Chan et al., 2022). Bertsimas & King (2016) formulated a mixed-integer quadratic optimization (MIQO) model to construct linear regression models that have various desirable properties, such as avoiding multicollinearity. Bertsimas & King (2017) extended this approach to logistic regression models. The condition number and the variance inflation factor (VIF) are frequently-used indicators for detecting multicollinearity in explanatory variables (Chatterjee & Hadi, 2015). Tamura et al. (2017) developed a MIQO-based cutting-plane algorithm that iteratively eliminates collinear subsets of explanatory variables based on the condition number. Tamura et al. (2019) proposed MIQO formulations for best subset selection with the VIF constraints on selected explanatory variables. Bertsimas & Li (2020) explored MIQO-based algorithm for best subset selection considering statistical significance and multicollinearity. The statistical significance of regression coefficients has also been addressed in several other papers (Carrizosa et al., 2020; Chung et al., 2020; Yarahmadi et al., 2024). However, these subset selection methods considering multicollinearity are computationally inefficient and difficult to apply to large-scale problems.

1.3 Our contribution

The goal of this paper is to establish a high-performance method for selecting the best subset of explanatory variables without multicollinearity for linear and logistic regression models. For this purpose, we first derive a bilevel reformulation of the optimization problem for best subset selection. We then design a two-way cutting-plane algorithm that uses cutting planes in two ways; one type of cutting planes based on Bertsimas et al. (2020; 2021) is used to approximate the upper-level nonlinear objective function, and the other type of cutting planes based on Tamura et al. (2017) is used to remove collinear subsets of explanatory variables. Our method is applicable to both linear and logistic regression models and can use any indicators (e.g., the

condition number and the VIF) for detecting multicollinearity. We also prove that our algorithm outputs a solution with guaranteed global optimality within a finite number of iterations.

We evaluate the effectiveness of our method through computational experiments using synthetic and public datasets for regression and binary classification. We compare the performance of our method with the L_1-regularized estimation (Friedman et al., 2010) and the previous cutting-plane algorithm (Tamura et al., 2017) for eliminating multicollinearity. Computational results demonstrate that our method can always find solutions that satisfy the multicollinearity constraint, and those solutions frequently deliver better predictive performance than do the other methods. Moreover, by increasing the weight of the L_2-regularization term in the objective function, we can significantly accelerate our method while maintaining high predictive performance.

1.4 Notation

Throughout this paper, we denote the set of consecutive integers as $[n] \coloneqq \{1, 2, \ldots, n\}$. We write a p-dimensional column vector as $\boldsymbol{a} \coloneqq (a_j)_{j\in[p]} \in \mathbb{R}^p$, and an $n \times p$ matrix as $\boldsymbol{X} \coloneqq (x_{ij})_{(i,j)\in[n]\times[p]} \in \mathbb{R}^{n\times p}$. The Hadamard product of vectors $\boldsymbol{a} \coloneqq (a_j)_{j\in[p]} \in \mathbb{R}^p$ and $\boldsymbol{b} \coloneqq (b_j)_{j\in[p]} \in \mathbb{R}^p$ is written as $\boldsymbol{a}\circ\boldsymbol{b} \coloneqq (a_j b_j)_{j\in[p]} \in \mathbb{R}^p$.

2 Regression models

In this section, we introduce the linear and logistic regression models to be considered for subset selection. Let us suppose that we are given a training dataset $\{(y_i, x_{i1}, x_{i2}, \ldots, x_{ip}) \mid i \in [n]\}$ consisting of n data instances. Here, y_i is the response variable to be predicted, and x_{ij} is the j-th explanatory variable for each instance $i \in [n]$.

The design matrix of explanatory variables is written as

$$\boldsymbol{X} \coloneqq (x_{ij})_{(i,j)\in[n]\times[p]} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix} = \begin{pmatrix} \boldsymbol{x}_{1\bullet}^\top \\ \boldsymbol{x}_{2\bullet}^\top \\ \vdots \\ \boldsymbol{x}_{n\bullet}^\top \end{pmatrix} = \begin{pmatrix} \boldsymbol{x}_{\bullet 1} & \boldsymbol{x}_{\bullet 2} & \cdots & \boldsymbol{x}_{\bullet p} \end{pmatrix} \in \mathbb{R}^{n\times p},$$

and the response vector is defined for each regression model as

$$\boldsymbol{y} \coloneqq (y_i)_{i\in[n]} \in \begin{cases} \mathbb{R}^n & \text{(linear regression)}, \\ \{-1, +1\}^n & \text{(logistic regression)}. \end{cases}$$

For notational simplicity, we assume that all explanatory variables are centered and scaled for unit length, and that the response variable is centered for linear regression; that is,

$$\sum_{i=1}^n x_{ij} = 0, \quad \sum_{i=1}^n (x_{ij})^2 = 1 \quad (j \in [p]), \tag{1}$$

$$\sum_{i=1}^n y_i = 0 \quad \text{(linear regression)}. \tag{2}$$

A vector of regression coefficients to be estimated is denoted as

$$\boldsymbol{a} \coloneqq (a_j)_{j\in[p]} \in \mathbb{R}^p.$$

2.1 Linear regression

The multiple linear regression model is formulated as

$$y_i = \boldsymbol{a}^\top \boldsymbol{x}_{i\bullet} + e_i = \sum_{j=1}^p a_j x_{ij} + e_i \quad (i \in [n]),$$

or equivalently,

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{a} + \boldsymbol{e} = \sum_{j=1}^{p} a_j \boldsymbol{x}_{\bullet j} + \boldsymbol{e},$$

where $\boldsymbol{e} \coloneqq (e_i)_{i\in[n]} \in \mathbb{R}^n$ is a vector composed of prediction residuals for each instance $i \in [n]$.

The ordinary least squares method estimates the regression coefficients by minimizing the residual sum of squares:

$$\sum_{i=1}^{n} (e_i)^2 = \sum_{i=1}^{n} (y_i - \boldsymbol{a}^\top \boldsymbol{x}_{i\bullet})^2 = (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{a})^\top (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{a}). \tag{3}$$

This amounts to solving the following convex quadratic optimization problem:

$$\text{minimize} \quad \frac{1}{2}\sum_{i=1}^{n} (y_i - \boldsymbol{a}^\top \boldsymbol{x}_{i\bullet})^2 \quad \text{subject to} \quad \boldsymbol{a} \in \mathbb{R}^p. \tag{4}$$

It is well known that by differentiating Eq. (3) with respect to $\boldsymbol{a}$, the optimality condition of problem (4) can be reduced to the normal equation:

$$\boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{a} = \boldsymbol{X}^\top \boldsymbol{y}. \tag{5}$$

2.2 Logistic regression

In the logistic regression model, the occurrence probability of binary response $y_i \in \{-1, +1\}$ is represented with a sigmoid function as

$$\Pr(\, y_i \mid \boldsymbol{x}_{i\bullet} \,) \coloneqq \frac{1}{1 + \exp\left(-y_i \boldsymbol{a}^\top \boldsymbol{x}_{i\bullet}\right)} \quad (i \in [n]).$$

The maximum likelihood method for estimating the regression coefficients maximizes the log-likelihood function:

$$\log \prod_{i=1}^{n} \Pr(\, y_i \mid \boldsymbol{x}_{i\bullet} \,) = \sum_{i=1}^{n} \log \Pr(\, y_i \mid \boldsymbol{x}_{i\bullet} \,) = -\sum_{i=1}^{n} \log\left(1 + \exp\left(-y_i \boldsymbol{a}^\top \boldsymbol{x}_{i\bullet}\right)\right). \tag{6}$$

After reversing the sign of the log-likelihood function (6), the maximum likelihood estimation is reduced to the following nonlinear convex optimization problem:

$$\text{minimize} \quad \sum_{i=1}^{n} \log\left(1 + \exp\left(-y_i \boldsymbol{a}^\top \boldsymbol{x}_{i\bullet}\right)\right) \quad \text{subject to} \quad \boldsymbol{a} \in \mathbb{R}^p. \tag{7}$$

3 Indicators for detecting multicollinearity

In this section, we introduce two indicators for detecting multicollinearity; namely, the condition number of the correlation matrix of explanatory variables and the variance inflation factor (VIF) for each explanatory variable. Here, we denote by $S \subseteq [p]$ a subset of explanatory variables to be explored.

3.1 Condition number

Since the explanatory variables are normalized as in Eq. (1), a subset correlation matrix of explanatory variables can be defined as

$$\boldsymbol{R}_S \coloneqq (r_{jk})_{(j,k)\in S\times S} \coloneqq \boldsymbol{X}_S^\top \boldsymbol{X}_S \in \mathbb{R}^{|S|\times|S|},$$

where $\boldsymbol{X}_S \coloneqq (\boldsymbol{x}_{\bullet j})_{j \in S} \in \mathbb{R}^{n \times |S|}$ is the design matrix corresponding to the subset S. The condition number of the subset correlation matrix is then expressed as

$$\mathrm{CN}(\boldsymbol{R}_S) \coloneqq \frac{\lambda_{\max}(\boldsymbol{R}_S)}{\lambda_{\min}(\boldsymbol{R}_S)}, \tag{8}$$

where $\lambda_{\min}(\boldsymbol{R}_S)$ and $\lambda_{\max}(\boldsymbol{R}_S)$ are the minimum and maximum eigenvalues of matrix $\boldsymbol{R}_S$.

When $\mathrm{CN}(\boldsymbol{R}_S) = \mathrm{CN}(\boldsymbol{X}_S^\top \boldsymbol{X}_S)$ is very large, the coefficient matrix of the corresponding normal equation (5) is ill-conditioned, thus implying that the regression coefficient estimates are numerically unstable. It is known that values of the condition number above 225 result in strong adverse effects of multicollinearity (Chatterjee & Hadi, 2015).

3.2 Variance inflation factor

Let us consider the following subset linear regression model for expressing the j-th explanatory variable in terms of the other explanatory variables:

$$\boldsymbol{x}_{\bullet j} = \sum_{k \in S \setminus \{j\}} a_k^{(j,S)} \boldsymbol{x}_{\bullet k} + \boldsymbol{e}^{(j,S)} \quad (j \in S), \tag{9}$$

where $\boldsymbol{a}^{(j,S)} \coloneqq (a_k^{(j,S)})_{k \in S \setminus \{j\}} \in \mathbb{R}^{|S|-1}$ and $\boldsymbol{e}^{(j,S)} \coloneqq (e_i^{(j,S)})_{i \in [n]} \in \mathbb{R}^n$ are vectors of regression coefficients and prediction residuals, respectively.

The goodness-of-fit of the subset regression model (9) can be measured by the coefficient of determination:

$$R^2(j,S) \coloneqq 1 - \frac{\sum_{i=1}^n (e_i^{(j,S)})^2}{\sum_{i=1}^n (x_{ij} - \bar{x}_j)^2} \quad (j \in S),$$

where $\bar{x}_j \coloneqq (\sum_{i=1}^n x_{ij})/n$ is the sample mean of the j-th explanatory variable. Then, VIF of the j-th explanatory variable is defined as

$$\mathrm{VIF}(j,S) \coloneqq \frac{1}{1 - R^2(j,S)} \quad (j \in S). \tag{10}$$

When $\mathrm{VIF}(j,S)$ is very large, $R^2(j,S)$ is close to one, which means that the j-th explanatory variable has a strong linear relationship with the other explanatory variables. As a rule of thumb, VIF values above 5 or 10 indicate problematic levels of multicollinearity (James et al., 2014).

4 Two-way cutting-plane algorithm

In this section, we first formulate an optimization problem to select the best subset of explanatory variables for regression models with the multicollinearity constraint. We then describe our two-way cutting-plane algorithm based on bilevel reformulation for solving the subset selection problem. We also prove the finite convergence of our algorithm.

4.1 Mixed-integer optimization formulation

Let $\boldsymbol{z} \coloneqq (z_j)_{j \in [p]} \in \{0,1\}^p$ be a vector composed of binary decision variables for subset selection; namely, $z_j = 1$ if the j-th variable is selected, and $z_j = 0$ otherwise. We then define the unified multicollinearity function from Eqs. (8) and (10) as

$$\mathrm{MC}(\boldsymbol{z}) \coloneqq \begin{cases} \mathrm{CN}(\boldsymbol{R}_{S(\boldsymbol{z})}) & \text{(condition number)}, \\ \max_{j \in S(\boldsymbol{z})} \mathrm{VIF}(j, S(\boldsymbol{z})) & \text{(variance inflation factor)}, \end{cases} \tag{11}$$

where $S(\boldsymbol{z}) \coloneqq \{ j \in [p] \mid z_j = 1 \}$ is the subset of explanatory variables specified by $\boldsymbol{z}$.

We next define the unified loss function from Eqs. (4) and (7) to be minimized for model estimation as

$$\ell(y,u) \coloneqq \begin{cases} \frac{1}{2}(y-u)^2 & \text{(linear regression)}, \\ \log(1+\exp(-yu)) & \text{(logistic regression)}, \end{cases} \tag{12}$$

where y and u correspond to observed and predicted responses, respectively.

We also introduce user-defined parameters to be employed for subset selection. Let

- $\gamma \in \mathbb{R}_+$ be a regularization parameter for adjusting the weight of the L_2-regularization term,
- $\alpha \in \mathbb{R}_+$ be a multicollinearity parameter for limiting the degree of multicollinearity, and
- $\theta \in \mathbb{Z}_+$ be a cardinality parameter for specifying the subset size of selected explanatory variables.

We are now in a position to formulate our subset selection problem:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{n} \ell(y_i, \boldsymbol{a}^\top \boldsymbol{x}_{i\bullet}) + \frac{1}{2\gamma}\boldsymbol{a}^\top \boldsymbol{a} && \text{(13a)} \\ \text{subject to} \quad & z_j = 0 \;\Rightarrow\; a_j = 0 \quad (j \in [p]), && \text{(13b)} \\ & \mathrm{MC}(\boldsymbol{z}) \le \alpha, && \text{(13c)} \\ & \sum_{j=1}^{p} z_j \le \theta, && \text{(13d)} \\ & \boldsymbol{a} \in \mathbb{R}^p,\ \boldsymbol{z} \in \{0,1\}^p. && \text{(13e)} \end{aligned}$$

Here, Eq. (13a) is the sum of the loss function (12) and the L_2-regularization term. Eq. (13b) is the logical implication for subset selection ensuring that the j-th regression coefficient must be zero when $z_j = 0$. Eq. (13c) is the upper-bound constraint on the degree of multicollinearity quantified by Eq. (11). Eq. (13d) is the cardinality constraint for limiting the subset size of explanatory variables. Eq. (13e) lists all of the decision variables.

Note that the previous formulation in Tamura et al. (2017) for eliminating multicollinearity does not contain the L_2-regularization term. In contrast, we add the L_2-regularization term in Eq. (13a), which makes it possible to develop our two-way cutting-plane algorithm. Note also that the importance of the L_2-regularization term in subset selection was pointed out by Mazumder et al. (2023).

4.2 Bilevel reformulation

We will now present bilevel reformulation of the subset selection problem (13) by following Bertsimas et al. (2020; 2021). Specifically, we decompose the optimization problem (13) into the *upper-level problem* for subset selection:

$$\begin{aligned} \text{minimize} \quad & f(\boldsymbol{z}) && \text{(14a)} \\ \text{subject to} \quad & \mathrm{MC}(\boldsymbol{z}) \le \alpha, && \text{(14b)} \\ & \sum_{j=1}^{p} z_j \le \theta, && \text{(14c)} \\ & \boldsymbol{z} \in \{0,1\}^p, && \text{(14d)} \end{aligned}$$

and the *lower-level problem* for calculating the upper-level objective function:

$$\begin{aligned} f(\boldsymbol{z}) \coloneqq \text{minimize} \quad & \sum_{i=1}^{n} \ell(y_i, (\boldsymbol{z} \circ \boldsymbol{a})^\top \boldsymbol{x}_{i\bullet}) + \frac{1}{2\gamma}\boldsymbol{a}^\top \boldsymbol{a} && \text{(15a)} \\ \text{subject to} \quad & \boldsymbol{a} \in \mathbb{R}^p, && \text{(15b)} \end{aligned}$$

where the logical implication (13b) is imposed by $\boldsymbol{z} \circ \boldsymbol{a}$ in Eq. (15a); this validity is ensured by the L_2-regularization term.

Note that strong duality holds for the lower-level problem (15) due to Slater's condition. Therefore, the objective function $f(\boldsymbol{z})$ can also be defined by the dual form of the lower-level problem (15):

$$f(\boldsymbol{z}) \coloneqq \text{maximize} \quad -\sum_{i=1}^{n} \hat{\ell}(y_i, u_i) - \frac{\gamma}{2} \sum_{j=1}^{p} z_j \left(\boldsymbol{u}^\top \boldsymbol{x}_{\bullet j} \boldsymbol{x}_{\bullet j}^\top \boldsymbol{u} \right) \tag{16a}$$

$$\text{subject to} \quad \boldsymbol{u} \in \mathbb{R}^n, \tag{16b}$$

where $\boldsymbol{u} \coloneqq (u_i)_{i \in [n]} \in \mathbb{R}^n$ is a vector of dual decision variables, and $\hat{\ell}(y, u)$ is the Fenchel conjugate of the loss function (12) defined as

$$\hat{\ell}(y, u) \coloneqq \begin{cases} \frac{1}{2} u^2 + yu & \text{(linear regression)}, \\ (1 + yu) \log(1 + yu) - yu \log(-yu) & \text{(logistic regression)}. \end{cases}$$

Let $\boldsymbol{a}^\star(\boldsymbol{z}) \in \mathbb{R}^p$ be an optimal solution to the primal lower-level problem (15), and $\boldsymbol{u}^\star(\boldsymbol{z}) \in \mathbb{R}^n$ be an optimal solution to the dual lower-level problem (16). As pointed out in Bertsimas et al. (2020; 2021), the dual optimal solution can be derived using the optimality condition from the primal optimal solution as

$$\boldsymbol{u}^\star(\boldsymbol{z}) = \begin{cases} -\boldsymbol{y} + \boldsymbol{X} \boldsymbol{a}^\star(\boldsymbol{z}) & \text{(linear regression)}, \\ \left(\frac{-y_i}{1 + \exp(y_i \boldsymbol{a}^\star(\boldsymbol{z})^\top \boldsymbol{x}_{i\bullet})} \right)_{i \in [n]} & \text{(logistic regression)}. \end{cases} \tag{17}$$

The primal lower-level problem (15) corresponding to regression with at most θ explanatory variables can be solved efficiently using sophisticated algorithms (e.g., Friedman et al. (2010)) implemented in modern machine learning libraries. Therefore, the dual lower-level problem (16) can also be solved efficiently through Eq. (17).

It is known that $f(\boldsymbol{z})$ is a convex function in $\boldsymbol{z} \in [0, 1]^p$ because it is defined by Eq. (16a) as the pointwise maximum of linear functions (cf. Section 3.2.3 in Boyd & Vandenberghe (2004)). In addition, the subgradient of $f(\boldsymbol{z})$ is derived through Eq. (16a) as

$$\begin{aligned} \boldsymbol{g}(\boldsymbol{z}) &\coloneqq \left(-\frac{\gamma}{2} \boldsymbol{u}^\star(\boldsymbol{z})^\top \boldsymbol{x}_{\bullet j} \boldsymbol{x}_{\bullet j}^\top \boldsymbol{u}^\star(\boldsymbol{z}) \right)_{j \in [p]} \\ &= -\frac{\gamma}{2} (\boldsymbol{u}^\star(\boldsymbol{z})^\top \boldsymbol{X}) \circ (\boldsymbol{u}^\star(\boldsymbol{z})^\top \boldsymbol{X}) \in \partial f(\boldsymbol{z}) \subseteq \mathbb{R}^p. \end{aligned} \tag{18}$$

4.3 Algorithm description

We will now formulate our two-way cutting-plane algorithm, which uses cutting planes in two ways to solve the upper-level problem (14). One type of cutting planes based on Bertsimas et al. (2020; 2021) is used to approximate the objective function (14a), and the other type of cutting planes based on Tamura et al. (2017) is used to remove solutions that violate the multicollinearity constraint (14b).

Let $\eta \in \mathbb{R}$ be a decision variable for representing the objective value of the upper-level problem (14). We first remove the multicollinearity constraint (14b) from the upper-level problem (14), and approximate the objective function (14a) from below by a tangent plane at $\boldsymbol{z}^{(0)} \in \{0, 1\}^p$ based on Eq. (18).

Specifically, we focus on the following relaxed upper-level problem:

$$\text{minimize} \quad \eta \tag{19a}$$

$$\text{subject to} \quad (\eta, \boldsymbol{z}) \in \mathcal{F}, \tag{19b}$$

where the feasible region is defined as

$$\mathcal{F} \leftarrow \left\{ (\eta, \boldsymbol{z}) \in \mathbb{R} \times \{0,1\}^p \;\middle|\; \sum_{j=1}^{p} z_j \leq \theta, \quad \eta \geq f(\boldsymbol{z}^{(0)}) + \boldsymbol{g}(\boldsymbol{z}^{(0)})^\top (\boldsymbol{z} - \boldsymbol{z}^{(0)}) \right\}. \tag{20}$$

We set the iteration count $t \leftarrow 1$, and let $(\eta^{(t)}, \boldsymbol{z}^{(t)})$ be an optimal solution to the relaxed problem (19).

Let $f^\star$ be the optimal objective value of problem (14). Then, $\boldsymbol{z}^{(t)}$ is called an *ε-optimal solution* to problem (14) if it is a feasible solution to the problem (14) and satisfies the following condition:

$$f(\boldsymbol{z}^{(t)}) \leq f^\star + \varepsilon,$$

where $\varepsilon \in \mathbb{R}_+$ is a threshold parameter for optimality. Since the relaxed problem (19) ensures that $\eta^{(t)} \leq f^\star$, we consider the following termination condition:

$$f(\boldsymbol{z}^{(t)}) \leq \eta^{(t)} + \varepsilon \quad \text{and} \quad \mathrm{MC}(\boldsymbol{z}^{(t)}) \leq \alpha.$$

If $f(\boldsymbol{z}^{(t)}) > \eta^{(t)} + \varepsilon$, then the approximation accuracy of the objective function is insufficient. In this case, to improve the approximation accuracy at $\boldsymbol{z}^{(t)}$, we update the feasible region as

$$\mathcal{F} \leftarrow \mathcal{F} \cap \left\{ (\eta, \boldsymbol{z}) \in \mathbb{R} \times \{0,1\}^p \;\middle|\; \eta \geq f(\boldsymbol{z}^{(t)}) + \boldsymbol{g}(\boldsymbol{z}^{(t)})^\top (\boldsymbol{z} - \boldsymbol{z}^{(t)}) \right\}. \tag{21}$$

If $\mathrm{MC}(\boldsymbol{z}^{(t)}) > \alpha$, then multicollinearity exists in the selected subset of explanatory variables. In this case, we first search for $\bar{\boldsymbol{z}}^{(t)}$ corresponding to a smaller subset of collinear variables (i.e., $\bar{\boldsymbol{z}}^{(t)} \leq \boldsymbol{z}^{(t)}$ and $\mathrm{MC}(\bar{\boldsymbol{z}}^{(t)}) > \alpha$) by executing the backward elimination method (Tamura et al., 2017) from $\boldsymbol{z}^{(t)}$. We then update the feasible region to remove $\bar{\boldsymbol{z}}^{(t)}$ as

$$\mathcal{F} \leftarrow \mathcal{F} \cap \left\{ (\eta, \boldsymbol{z}) \in \mathbb{R} \times \{0,1\}^p \;\middle|\; (\bar{\boldsymbol{z}}^{(t)})^\top \boldsymbol{z} \leq (\bar{\boldsymbol{z}}^{(t)})^\top (\bar{\boldsymbol{z}}^{(t)}) - 1 \right\}. \tag{22}$$

After that, we solve the relaxed upper-level problem (19) again with the updated feasible region. Our two-way cutting-plane algorithm is summarized in Algorithm 1.

Algorithm 1 Two-way cutting-plane algorithm for solving problem (14)

Input: L_2-regularization parameter $\gamma \in \mathbb{R}_+$, multicollinearity parameter $\alpha \in \mathbb{R}_+$, cardinality parameter $\theta \in \mathbb{Z}_+$, optimality parameter $\varepsilon \in \mathbb{R}_+$, and initial solution $\boldsymbol{z}^{(0)} \in \{0,1\}^p$.

Set the iteration count $t \leftarrow 1$ and the feasible region $\mathcal{F}$ as in Eq. (20).
repeat
  Compute an optimal solution $(\eta^{(t)}, \boldsymbol{z}^{(t)})$ to the relaxed problem (19).
  if $f(\boldsymbol{z}^{(t)}) > \eta^{(t)} + \varepsilon$ **then**
    Solve the lower-level problem (15) to calculate an optimal solution $\boldsymbol{a}^\star(\boldsymbol{z}^{(t)})$ and the associated objective value $f(\boldsymbol{z}^{(t)})$.
    Compute the subgradient $\boldsymbol{g}(\boldsymbol{z}^{(t)})$ based on Eqs. (17) and (18).
    Update the feasible region $\mathcal{F}$ as in Eq. (21).
  end if
  if $\mathrm{MC}(\boldsymbol{z}^{(t)}) > \alpha$ **then**
    Compute $\bar{\boldsymbol{z}}^{(t)}$ such that $\bar{\boldsymbol{z}}^{(t)} \leq \boldsymbol{z}^{(t)}$ and $\mathrm{MC}(\bar{\boldsymbol{z}}^{(t)}) > \alpha$ using the backward elimination method (Tamura et al., 2017).
    Update the feasible region $\mathcal{F}$ as in Eq. (22).
  end if
  Set $t \leftarrow t + 1$.
until $f(\boldsymbol{z}^{(t)}) \leq \eta^{(t)} + \varepsilon$ and $\mathrm{MC}(\boldsymbol{z}^{(t)}) \leq \alpha$.

Output: ε-optimal solution $\boldsymbol{z}^{(t)} \in \{0,1\}^p$ to problem (14).

4.4 Convergence property

We will now prove the finite convergence of our two-way cutting-plane algorithm by following Kobayashi et al. (2023) and Tamura et al. (2017).

Theorem 1. *Algorithm 1 terminates in a finite number of iterations and then outputs an ε-optimal solution to problem* (14)*.*

Proof. Since the number of solutions of $\boldsymbol{z} \in \{0,1\}^p$ is finite, Algorithm 1 must generate the same solution more than once in a sufficiently large but finite number of iterations. Therefore, we can suppose that Algorithm 1 generates $\tilde{\boldsymbol{z}} \in \{0,1\}^p$ at the t_1-th and t_2-th iterations ($t_1 < t_2$), that is, $\boldsymbol{z}^{(t_1)} = \boldsymbol{z}^{(t_2)} = \tilde{\boldsymbol{z}}$. In what follows, we will show that $\mathrm{MC}(\tilde{\boldsymbol{z}}) \leq \alpha$ and $f(\tilde{\boldsymbol{z}}) \leq \eta^{(t_2)} + \varepsilon$, which implies that the algorithm terminates with an ε-optimal solution in at most t_2 iterations.

First, we prove that $\mathrm{MC}(\tilde{\boldsymbol{z}}) \leq \alpha$ by contradiction. Accordingly, we suppose that $\mathrm{MC}(\tilde{\boldsymbol{z}}) > \alpha$. In this case, $\boldsymbol{z}^{(t_1)} = \tilde{\boldsymbol{z}}$ is removed from the feasible region by Eq. (22) at the t_1-th iteration, which contradicts that the algorithm generates $\boldsymbol{z}^{(t_2)} = \tilde{\boldsymbol{z}}$ at the t_2-th iteration. Consequently, we have $\mathrm{MC}(\tilde{\boldsymbol{z}}) \leq \alpha$ (i.e., $\tilde{\boldsymbol{z}}$ is a feasible solution to problem (14)).

Next, we prove that $f(\tilde{\boldsymbol{z}}) \leq \eta^{(t_2)} + \varepsilon$. If $f(\tilde{\boldsymbol{z}}) \leq \eta^{(t_1)} + \varepsilon$, then the algorithm terminates at the t_1-th iteration. Otherwise, the feasible region $\mathcal{F}$ is updated by Eq. (21) at the t_1-th iteration. Since $(\eta^{(t_2)}, \boldsymbol{z}^{(t_2)}) \in \mathcal{F}$ at the t_2-th iteration, it follows that

$$f^\star \geq \eta^{(t_2)} \geq f(\boldsymbol{z}^{(t_1)}) + \boldsymbol{g}(\boldsymbol{z}^{(t_1)})^\top (\boldsymbol{z}^{(t_2)} - \boldsymbol{z}^{(t_1)}) = f(\tilde{\boldsymbol{z}}) \geq f^\star.$$

This implies that

$$f(\tilde{\boldsymbol{z}}) = f^\star = \eta^{(t_2)} \leq \eta^{(t_2)} + \varepsilon,$$

which completes the proof. □

5 Computational experiments

In this section, we evaluate the effectiveness of our method for best subset selection considering multicollinearity through computational experiments using synthetic and public datasets. All computations were performed on a Windows computer equipped with an Intel Core i7-8700 CPU (3.20 GHz) and 48 GB of memory.

5.1 Experimental setup

We compare the performance of the following subset selection methods:

- L1-Reg: L_1-regularized estimation (Friedman et al., 2010);
- $\mathrm{CPA_{MC}}$: cutting-plane algorithm (Tamura et al., 2017) for eliminating multicollinearity based on Eq. (22);
- $\mathrm{CPA_{2W}}$: our two-way cutting-plane algorithm (Algorithm 1).

We implemented the L_1-regularized estimation using the Python scikit-learn library, where the L_1-regularization parameter was tuned by the binary search such that the number of nonzero regression coefficients was equal to the cardinality parameter θ. We implemented the cutting-plane algorithms in Python, where the relaxed upper-level problem (19) was solved using CPLEX[1] 22.1.1.0, and the lower-level problem (15) was solved using the Python scikit-learn library. We also used the lazy constraint callback to add cutting planes (21) and (22) during a branch-and-bound procedure. In the $\mathrm{CPA_{2W}}$ method, we simply set $z_j^{(0)} = 1$ ($j \leq \theta$) and $z_j^{(0)} = 0$ ($j \geq \theta + 1$) as the initial solution. In the $\mathrm{CPA_{MC}}$ method for logistic regression,

[1] `https://www.ibm.com/products/ilog-cplex-optimization-studio`

the nonlinear objective function (7) was linearized using a piecewise-linear approximation (Sato et al., 2016) with 17 tangent lines.

In the $\mathrm{CPA}_{2\mathrm{W}}$ method, the L_2-regularization parameter was set by following Bertsimas et al. (2020) as

$$\gamma = \frac{2^m \cdot p}{n \cdot \theta \cdot \max_{i \in [n]} \|\boldsymbol{x}_{i\bullet}\|_2^2}. \tag{23}$$

Here, the regularization exponent parameter $m \in \mathbb{R}$ was tuned according to the standard-error rule (James et al., 2014) based on 10-fold cross-validation with the training dataset. Specifically, we first executed the two-way cutting-plane algorithm for $m \in \{0, 2, \ldots, 18\}$ with a time limit of 30 s. Next, for each $m \in \{0, 2, \ldots, 18\}$ we performed 10-fold cross-validation of the L_2-regularized estimation with the selected subset of explanatory variables, where the cross-validation error was evaluated by the residual sum of squares for linear regression and by the negative log-likelihood for logistic regression. Let $\mathrm{CV}(m)$ and $\mathrm{Std}(m)$ be respectively the mean and standard deviation of the cross-validation error for each m. Then, we chose the smallest $m \in \{0, 2, \ldots, 18\}$ such that $\mathrm{CV}(m) \leq \mathrm{CV}(m^\star) + k \cdot \mathrm{Std}(m^\star)$ for each $k \in \{0.0, 0.5, 1.5, 2.0\}$, where $m^\star \in \arg\min_{m \in \{0,2,\ldots,18\}} \mathrm{CV}(m)$. Note that as k increases, the weight of the L_2-regularization term increases in the objective function (13a), so k can be regarded as a regularization strength parameter.

We used the root mean square error (RMSE) for linear regression and the area under the ROC curve (AUC) for logistic regression as evaluation metrics of predictive performance. Subset selection was performed using training datasets to estimate subset regression models, and their predictive performance was evaluated using testing datasets. In the tables of experimental results, the best testing RMSE/AUC values for each problem configuration are given in bold. We set $\theta = 10$ for the cardinality parameter. We used VIF as the indicator for detecting multicollinearity, where the corresponding upper-bound parameter was set as $\alpha = 10$. We also evaluate the following metrics:

- $|S|$: selected subset size of explanatory variables (i.e., the number of nonzero regression coefficients);
- #Cuts (Obj): number of cutting planes added by Eq. (21) to approximate the objective function;
- #Cuts (MC): number of cutting planes added by Eq. (22) to avoid multicollinearity;
- Gap: relative optimality gap as a percentage of the objective function (13a);
- Time: computation time in seconds required to select a subset of explanatory variables.

A subset selection computation was terminated if it did not complete within 180 s; in those cases, the best feasible solution found within 180 s was taken as the result, along with the number of trials that reached the time limit.

5.2 Generation of synthetic datasets

We prepared synthetic datasets by following Bertsimas et al. (2020); Hastie et al. (2020). First, we specified a ground-truth vector $\boldsymbol{a}^\star \coloneqq (a_j^\star)_{j \in [p]} \in \mathbb{R}^p$ of regression coefficients, where $a_j^\star$ was uniformly drawn from $\{-1, 0, +1\}$ for each $j \in [p]$. Next, we sampled explanatory variables from a multivariate normal distribution as $\boldsymbol{x}_{i\bullet} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$ for $i \in [n]$, where the covariance matrix was defined as $\boldsymbol{\Sigma} \coloneqq (0.95^{|i-j|})_{(i,j) \in [p] \times [p]} \in \mathbb{R}^{p \times p}$.

Then, we sampled a prediction residual as $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$ for $i \in [n]$, where the standard deviation $\sigma \in \mathbb{R}_+$ was determined depending on the signal-to-noise ratio (SNR) as

$$\sigma^2 = \frac{\mathrm{Var}(\boldsymbol{X}\boldsymbol{a}^\star)}{\mathrm{SNR}} = \frac{(\boldsymbol{a}^\star)^\top \boldsymbol{\Sigma} \boldsymbol{a}^\star}{\mathrm{SNR}}. \tag{24}$$

Note here that when the SNR is large, the variance of the prediction residual is small.

Finally, we produced the response variable as

$$y_i = \begin{cases} (\boldsymbol{a}^\star)^\top \boldsymbol{x}_{i\bullet} + \varepsilon_i & \text{(linear regression)} \\ \mathrm{sign}((\boldsymbol{a}^\star)^\top \boldsymbol{x}_{i\bullet} + \varepsilon_i) & \text{(logistic regression)} \end{cases} \quad (i \in [n]),$$

Table 1: Results of linear regression ($\theta = 10$) for synthetic datasets

(n, p, SNR)	Method	k	m	Testing RMSE	VIF	$\lvert S \rvert$	#Cuts Obj	#Cuts MC	Gap [%]	Time [s]	
$(500, 50, 3.0)$	L1-Reg	NA	NA	0.6388 ± 0.027	16.6	10.0	NA	NA	NA	0.01	
	$\mathrm{CPA_{MC}}$	NA	NA	0.5443 ± 0.002	8.1	10.0	NA	9.6	6.35	> 180	(5/5)
	$\mathrm{CPA_{2W}}$	0.0	14.0	$\mathbf{0.5436} \pm 0.001$	8.1	10.0	6094.8	316.6	843.92	> 180	(4/5)
		0.5	8.4	0.5472 ± 0.003	8.8	10.0	1159.2	80.8	1.30	> 180	(1/5)
		1.0	6.4	0.5695 ± 0.004	9.4	10.0	295.8	57.2	< 0.01	12.16	
		2.0	5.6	0.6017 ± 0.017	9.4	10.0	273.6	53.0	< 0.01	13.18	
$(\mathbf{1000}, 50, 3.0)$	L1-Reg	NA	NA	0.6993 ± 0.020	17.4	10.0	NA	NA	NA	0.02	
	$\mathrm{CPA_{MC}}$	NA	NA	0.5598 ± 0.023	7.9	10.0	NA	6.0	11.74	> 180	(5/5)
	$\mathrm{CPA_{2W}}$	0.0	16.0	$\mathbf{0.5554} \pm 0.021$	7.8	10.0	7396.6	271.0	1273.94	> 180	(5/5)
		0.5	8.8	0.5635 ± 0.021	8.9	10.0	897.4	61.4	0.61	> 180	(1/5)
		1.0	8.4	0.5672 ± 0.020	9.2	10.0	872.6	58.8	0.64	> 180	(1/5)
		2.0	7.6	0.5814 ± 0.027	9.2	10.0	502.6	48.2	0.60	> 180	(1/5)
$(500, \mathbf{100}, 3.0)$	L1-Reg	NA	NA	0.7597 ± 0.022	16.2	10.0	NA	NA	NA	0.03	
	$\mathrm{CPA_{MC}}$	NA	NA	0.5901 ± 0.014	5.1	10.0	NA	4.4	29.15	> 180	(5/5)
	$\mathrm{CPA_{2W}}$	0.0	11.2	$\mathbf{0.5830} \pm 0.016$	5.2	10.0	3888.0	201.2	1553.69	> 180	(5/5)
		0.5	7.2	0.5971 ± 0.017	7.2	10.0	1361.2	75.8	4.78	> 180	(3/5)
		1.0	6.8	0.6032 ± 0.016	7.6	10.0	955.8	70.4	2.57	> 180	(2/5)
		2.0	5.2	0.6526 ± 0.019	8.7	10.0	344.0	56.6	< 0.01	19.29	
$(500, 50, \mathbf{1.0})$	L1-Reg	NA	NA	0.7544 ± 0.009	18.5	10.0	NA	NA	NA	0.02	
	$\mathrm{CPA_{MC}}$	NA	NA	0.7385 ± 0.002	7.5	10.0	NA	17.2	2.48	> 180	(5/5)
	$\mathrm{CPA_{2W}}$	0.0	14.0	$\mathbf{0.7363} \pm 0.002$	7.6	10.0	6341.4	394.0	203.23	> 180	(5/5)
		0.5	7.2	0.7375 ± 0.004	9.8	10.0	182.2	58.2	< 0.01	3.65	
		1.0	5.6	0.7581 ± 0.003	9.8	10.0	78.0	38.8	< 0.01	1.76	
		2.0	4.0	0.7995 ± 0.012	9.7	10.0	65.6	38.4	< 0.01	1.68	

where

$$\mathrm{sign}(x) := \begin{cases} +1 & \text{if } x \geq 0, \\ -1 & \text{otherwise.} \end{cases}$$

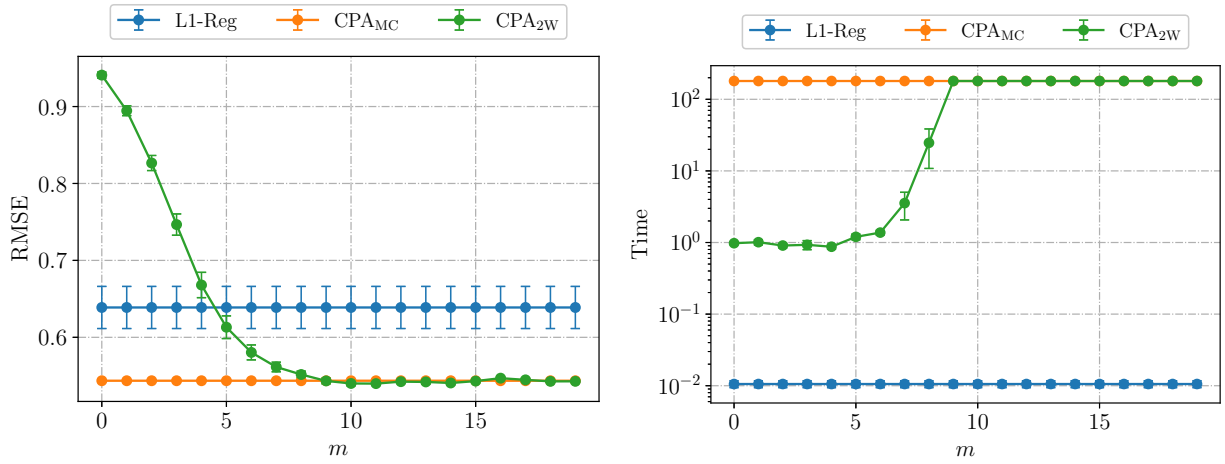
For linear regression, we standardized the response vector $\boldsymbol{y}$ to have a mean of 0 (cf. Eq. (2)) and a variance of 1. We generated training and testing datasets five times, and show the average of the five trials.

5.3 Results of linear regression for synthetic datasets

Table 1 gives the average results of five trials of linear regression on the synthetic datasets, where n is the number of training data instances, p is the number of candidate variables, and SNR is inversely proportional to the residual variance through Eq. (24). Standard errors are also given for the testing RMSE, which is the metric of predictive performance.

We begin by focusing on the results for the basic configuration $(n, p, \mathrm{SNR}) = (500, 50, 3.0)$. Although the L_1-regularized estimation (L1-Reg) had the shortest computation time, it failed to satisfy the multicollinearity constraint (i.e., $\mathrm{VIF} = 16.6 > \alpha = 10$) and provided the worst RMSE. The previous cutting-plane algorithm for eliminating multicollinearity ($\mathrm{CPA_{MC}}$) provided the second best RMSE while satisfying the multicollinearity constraint, but reached the 180 s time limit in all five trials. Our two-way cutting-plane algorithm ($\mathrm{CPA_{2W}}$) always satisfied the multicollinearity constraint and attained the best RMSE with $k = 0.0$. Moreover, the computation time of our $\mathrm{CPA_{2W}}$ method was reduced by increasing the regularization strength parameter k; in particular, a sufficiently good RMSE was obtained in 12.16 s when $k = 1.0$.

We next turn to the results for $(n, p, \mathrm{SNR}) = (\mathbf{1000}, 50, 3.0)$ when the number of training data instances is increased from $n = 500$ in the basic configuration to $n = 1000$. Compared to the results for the basic configuration, the optimality gap of the cutting-plane algorithms was often large, and their computations were



(a) Testing RMSE vs. m

(b) Subset selection time vs. m

Figure 1: Results of linear regression ($\theta = 10$) with respect to the regularization exponent parameter m for synthetic datasets with $(n, p, \mathrm{SNR}) = (500, 50, 3.0)$

frequently terminated by the 180 s time limit. However, our CPA$_{\mathrm{2W}}$ method with $k = 0.0$ attained the best RMSE while satisfying the multicollinearity constraint. Moreover, our CPA$_{\mathrm{2W}}$ method with $k \in \{0.5, 1.0\}$ completed the computations within the 180 s time limit in four out of five trials and also achieved sufficiently good RMSEs.

We then move on to the results for $(n, p, \mathrm{SNR}) = (500, \mathbf{100}, 3.0)$ when the number of candidate variables is increased from $p = 50$ in the basic configuration to $p = 100$. Even compared to the results of $n = 1000$, the optimality gap and computation time of the cutting-plane algorithms increased. Nevertheless, our CPA$_{\mathrm{2W}}$ method with $k = 0.0$ attained the best RMSE while satisfying the multicollinearity constraint. Moreover, increasing k for our CPA$_{\mathrm{2W}}$ method reduced the number of trials that reached the 180 s time limit.

We finally focus on the results for $(n, p, \mathrm{SNR}) = (500, 50, \mathbf{1.0})$ when the signal-to-noise ratio is decreased from SNR $= 3.0$ in the basic configuration to SNR $= 1.0$. Compared to other results, the difference in RMSE among the methods was small, and the computation time of our CPA$_{\mathrm{2W}}$ method was often short. Notably, our CPA$_{\mathrm{2W}}$ method with $k = 0.5$ achieved almost the best RMSE in only 3.65 s.

Figure 1 shows the testing RMSE and subset selection time of linear regression with respect to the regularization exponent parameter $m \in \{0, 1, \ldots, 19\}$ (see also Eq. (23)) for the synthetic datasets with $(n, p, \mathrm{SNR}) = (500, 50, 3.0)$, where the standard errors are represented by error bars. In Figure 1(a), our CPA$_{\mathrm{2W}}$ method reduced the RMSE by increasing the regularization exponent parameter m, outperforming the L1-Reg method from around $m = 5$, and achieving almost the same RMSE as the CPA$_{\mathrm{MC}}$ method from around $m = 9$. In Figure 1(b), the L1-Reg method had the shortest computation time, while the CPA$_{\mathrm{MC}}$ method always reached the 180 s time limit. In contrast, our CPA$_{\mathrm{2W}}$ method reduced the computation time by decreasing the regularization exponent parameter m. More importantly, our CPA$_{\mathrm{2W}}$ method with $m \in \{5, 6, 7, 8\}$ delivered better quality solutions than did the L1-Reg method (Figure 1(a)) within the 180 s time limit (Figure 1(b)).

5.4 Results of logistic regression for synthetic datasets

Table 2 gives the average results of five trials of logistic regression on the synthetic datasets, where n is the number of training data instances, p is the number of candidate variables, and SNR is inversely proportional to the residual variance through Eq. (24). Standard errors are also given for the testing AUC, which is the metric of predictive performance.

Table 2: Results of logistic regression ($\theta = 10$) for synthetic datasets

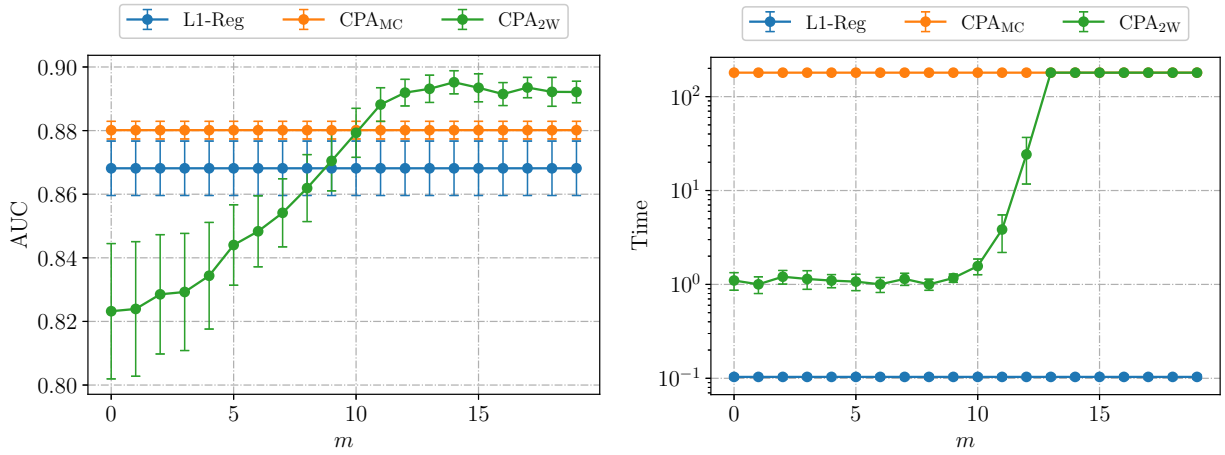
(n, p, SNR)	Method	k	m	Testing AUC	VIF	$\lvert S \rvert$	#Cuts Obj	#Cuts MC	Gap [%]	Time [s]	
$(500, 50, 3.0)$	L1-Reg	NA	NA	0.8682 ± 0.009	14.2	10.0	NA	NA	NA	0.10	
	$\mathrm{CPA_{MC}}$	NA	NA	0.8772 ± 0.004	8.8	10.0	NA	17.2	18.72	> 180	(5/5)
	$\mathrm{CPA_{2W}}$	0.0	15.6	$\mathbf{0.8958} \pm 0.004$	7.1	10.0	5152.8	267.6	63.30	> 180	(5/5)
		0.5	11.2	0.8894 ± 0.005	8.4	10.0	233.6	48.2	< 0.01	4.30	
		1.0	9.2	0.8727 ± 0.009	9.4	10.0	87.4	32.4	< 0.01	1.24	
		2.0	6.8	0.8596 ± 0.011	9.5	10.0	49.4	34.8	< 0.01	1.07	
$(\mathbf{1000}, 50, 3.0)$	L1-Reg	NA	NA	0.8838 ± 0.007	16.8	10.0	NA	NA	NA	0.17	
	$\mathrm{CPA_{MC}}$	NA	NA	0.7423 ± 0.050	4.7	4.6	NA	10.6	37.07	> 180	(5/5)
	$\mathrm{CPA_{2W}}$	0.0	15.2	$\mathbf{0.8996} \pm 0.005$	7.8	10.0	4310.2	189.4	29.66	> 180	(5/5)
		0.5	11.2	0.8973 ± 0.004	8.5	10.0	587.2	53.4	< 0.01	39.32	
		1.0	9.6	0.8920 ± 0.004	9.0	10.0	243.0	33.4	< 0.01	12.55	
		2.0	7.6	0.8776 ± 0.010	9.4	10.0	66.2	35.0	< 0.01	1.43	
$(500, \mathbf{100}, 3.0)$	L1-Reg	NA	NA	0.8250 ± 0.015	16.7	10.0	NA	NA	NA	0.32	
	$\mathrm{CPA_{MC}}$	NA	NA	0.7425 ± 0.052	4.0	5.8	NA	9.8	53.66	> 180	(5/5)
	$\mathrm{CPA_{2W}}$	0.0	14.8	0.8710 ± 0.007	5.1	10.0	4215.8	219.2	534.00	> 180	(5/5)
		0.5	10.4	$\mathbf{0.8741} \pm 0.009$	6.7	10.0	1776.4	70.4	3.61	> 180	(5/5)
		1.0	9.2	0.8713 ± 0.008	7.1	10.0	886.6	50.6	0.79	> 180	(4/5)
		2.0	6.8	0.8432 ± 0.013	9.0	10.0	107.6	34.8	< 0.01	2.28	
$(500, 50, \mathbf{1.0})$	L1-Reg	NA	NA	0.7951 ± 0.012	12.7	10.0	NA	NA	NA	0.09	
	$\mathrm{CPA_{MC}}$	NA	NA	0.7923 ± 0.009	8.0	9.8	NA	24.6	10.74	> 180	(5/5)
	$\mathrm{CPA_{2W}}$	0.0	14.0	0.8060 ± 0.008	8.2	10.0	3626.8	225.0	28.61	> 180	(3/5)
		0.5	9.2	$\mathbf{0.8073} \pm 0.008$	8.6	10.0	171.4	36.6	< 0.01	2.26	
		1.0	6.8	0.7927 ± 0.011	9.7	10.0	61.8	36.8	< 0.01	1.20	
		2.0	2.8	0.7660 ± 0.023	9.7	10.0	51.6	37.2	< 0.01	1.08	

We begin by focusing on the results for the basic configuration $(n, p, \mathrm{SNR}) = (500, 50, 3.0)$. As with the results of linear regression, the L_1-regularized estimation (L1-Reg) completed the computation in the shortest time, but it failed to satisfy the multicollinearity constraint (i.e., $\mathrm{VIF} = 14.2 > \alpha = 10$) and yielded a very poor AUC. The previous cutting-plane algorithm for eliminating multicollinearity ($\mathrm{CPA_{MC}}$) satisfied the multicollinearity constraint; however, it reached the 180 s time limit in all five trials and failed to produce a good AUC. In contrast, our two-way cutting-plane algorithm ($\mathrm{CPA_{2W}}$) with $k = 0.0$ achieved the best AUC while satisfying the multicollinearity constraint. Moreover, our $\mathrm{CPA_{2W}}$ method significantly reduced the computation time by increasing the regularization strength parameter k; in particular, it attained a much better AUC with $k = 0.5$ in only 4.30 s than did the L1-Reg and $\mathrm{CPA_{MC}}$ methods.

We next turn to the results for $(n, p, \mathrm{SNR}) = (\mathbf{1000}, 50, 3.0)$ when the number of training data instances is increased from $n = 500$ in the basic configuration to $n = 1000$. Compared to the results for the basic configuration, the AUC obtained by the $\mathrm{CPA_{MC}}$ method was significantly degraded, and its selected subset size was much smaller than $\theta = 10$. This is mainly because the $\mathrm{CPA_{MC}}$ method failed to find a good-quality solution within the 180 s time limit. In contrast, our $\mathrm{CPA_{2W}}$ method still attained very good AUCs. For example, our $\mathrm{CPA_{2W}}$ method with $k \in \{0.5, 1.0\}$ produced solutions of higher quality than the L1-Reg and $\mathrm{CPA_{MC}}$ solutions in a few tens of seconds.

We then move on to the results for $(n, p, \mathrm{SNR}) = (500, \mathbf{100}, 3.0)$ when the number of candidate variables is increased from $p = 50$ in the basic configuration to $p = 100$. Also in this case, the AUC obtained by the $\mathrm{CPA_{MC}}$ method was very poor. Compared to the results for the basic configuration, our $\mathrm{CPA_{2W}}$ method frequently reached the 180 s time limit. However, the solutions obtained by our $\mathrm{CPA_{2W}}$ method with $k \in \{0.0, 0.5, 1.0\}$ were much better than the L1-Reg and $\mathrm{CPA_{MC}}$ solutions.

We finally focus on the results for $(n, p, \mathrm{SNR}) = (500, 50, \mathbf{1.0})$ when the signal-to-noise ratio is decreased from $\mathrm{SNR} = 3.0$ in the basic configuration to $\mathrm{SNR} = 1.0$. Compared to the results for the basic configuration, the



(a) Testing AUC vs. m

(b) Subset selection time vs. m

Figure 2: Results of logistic regression ($\theta = 10$) with respect to the regularization exponent parameter m for synthetic datasets with $(n, p, \mathrm{SNR}) = (500, 50, 3.0)$

Table 3: Public datasets for regression and binary classification

Task	Name	n	p	Original dataset (Markelle et al., 2024)
Regression	`Crime`	1994	101	Communities and Crime
	`Student`	649	56	Student Performance
Binary classification	`SPECT`	267	44	SPECT Heart
	`Breast`	569	30	Breast Cancer Wisconsin (Diagnostic)

difference in AUC among the methods was small. Nevertheless, our CPA$_{\mathrm{2W}}$ method with $k = 0.5$ achieved the best AUC in only 2.26 s.

Figure 2 shows the testing AUC and subset selection time of logistic regression with respect to the regularization exponent parameter $m \in \{0, 1, \ldots, 19\}$ (see also Eq. (23)) for the synthetic datasets with $(n, p, \mathrm{SNR}) = (500, 50, 3.0)$, where the standard errors are represented by error bars. In Figure 2(a), the AUC of our CPA$_{\mathrm{2W}}$ method improved with the regularization exponent parameter m. As a result, our CPA$_{\mathrm{2W}}$ method outperformed the L1-Reg method when $m \geq 9$ and the CPA$_{\mathrm{MC}}$ method when $m \geq 11$. In Figure 2(b), while the L1-Reg method had the shortest computation time, the CPA$_{\mathrm{MC}}$ method was unable to finish the computation within the 180 s time limit. In contrast, our CPA$_{\mathrm{2W}}$ method reduced the computation time by decreasing the regularization exponent parameter m. Notably, our CPA$_{\mathrm{2W}}$ method with $m \in \{10, 11, 12\}$ found solutions comparable to or better than the L1-Reg and CPA$_{\mathrm{MC}}$ solutions (Figure 2(a)) within the 180 s time limit (Figure 2(b)).

5.5 Results for public datasets

Table 3 lists public datasets for regression and binary classification downloaded from the UCI Machine Learning Repository (Markelle et al., 2024), where n and p are the numbers of data instances and candidate variables, respectively. The experimental setup for these public datasets was the same as those for the synthetic datasets, and the predictive performance was evaluated using 5-fold cross-validation; that is, 80% of the data instances were used for training, and the remaining 20% were used for testing.

Table 4 gives results of linear regression on the public `Crime` and `Student` datasets. For the `Crime` dataset, the L_1-regularized estimation (L1-Reg) was much faster than the other methods and satisfied the multicollinearity constraint, whereas its RMSE was not very good. The previous cutting-plane algorithm for

Table 4: Results of linear regression ($\theta = 10$) for public datasets

Dataset	Method	k	m	Testing RMSE	VIF	$\lvert S\rvert$	#Cuts Obj	#Cuts MC	Gap [%]	Time [s]	
`Crime`	L1-Reg	NA	NA	0.6073 ± 0.008	8.0	10.0	NA	NA	NA	0.04	
	CPA_{MC}	NA	NA	0.6033 ± 0.008	7.0	10.0	NA	8.0	14.91	> 180	(5/5)
	CPA_{2W}	0.0	13.2	$\mathbf{0.5891} \pm 0.010$	6.7	10.0	5224.6	186.4	1093.18	> 180	(5/5)
		0.5	8.0	0.5982 ± 0.007	8.6	10.0	1674.0	196.0	0.04	> 180	(1/5)
		1.0	6.0	0.6305 ± 0.005	9.3	10.0	338.2	93.2	< 0.01	6.72	
		2.0	6.0	0.6305 ± 0.005	9.3	10.0	338.2	93.2	< 0.01	6.67	
`Student`	L1-Reg	NA	NA	0.8708 ± 0.008	1.7	10.0	NA	NA	NA	0.02	
	CPA_{MC}	NA	NA	$\mathbf{0.8648} \pm 0.013$	1.2	10.0	NA	0.0	3.27	> 180	(5/5)
	CPA_{2W}	0.0	11.6	0.8679 ± 0.016	1.2	10.0	6575.4	13.2	7106.96	> 180	(5/5)
		0.5	6.0	0.8660 ± 0.011	1.3	10.0	1317.6	8.2	0.23	> 180	(2/5)
		1.0	4.8	0.8818 ± 0.006	1.5	10.0	501.6	5.8	0.14	> 180	(1/5)
		2.0	3.2	0.9082 ± 0.009	1.8	10.0	49.4	3.8	< 0.01	0.38	

Table 5: Results of logistic regression ($\theta = 10$) for public datasets

Dataset	Method	k	m	Testing AUC	VIF	$\lvert S\rvert$	#Cuts Obj	#Cuts MC	Gap [%]	Time [s]	
`SPECT`	L1-Reg	NA	NA	0.7684 ± 0.027	$+\infty$	10.0	NA	NA	NA	0.02	
	CPA_{MC}	NA	NA	0.8005 ± 0.034	2.2	10.0	NA	0.0	0.38	> 180	(5/5)
	CPA_{2W}	0.0	8.8	0.8170 ± 0.024	2.2	10.0	182.0	15.8	0.03	> 180	(1/5)
		0.5	5.6	$\mathbf{0.8209} \pm 0.015$	2.2	10.0	34.8	11.4	< 0.01	0.32	
		1.0	2.4	0.8084 ± 0.012	2.4	10.0	19.0	10.2	< 0.01	0.19	
		2.0	0.0	0.8081 ± 0.015	2.3	10.0	13.0	9.2	< 0.01	0.15	
`Breast`	L1-Reg	NA	NA	0.9929 ± 0.004	57.2	10.0	NA	NA	NA	0.05	
	CPA_{MC}	NA	NA	0.9867 ± 0.003	8.0	10.0	NA	22.6	63.24	> 180	(5/5)
	CPA_{2W}	0.0	16.0	0.9945 ± 0.003	8.2	10.0	473.4	202.8	< 0.01	8.84	
		0.5	14.0	$\mathbf{0.9952} \pm 0.002$	8.6	10.0	310.0	186.2	< 0.01	5.47	
		1.0	12.0	0.9943 ± 0.002	9.1	9.4	216.4	175.2	< 0.01	4.11	
		2.0	10.4	0.9921 ± 0.002	9.4	9.2	229.0	194.0	< 0.01	4.43	

eliminating multicollinearity (CPA_{MC}) also satisfied the multicollinearity constraint, but it failed to complete the computation within the 180 s time limit in all five trials. Our two-way cutting-plane algorithm (CPA_{2W}) attained the best RMSE with $k = 0.0$, and it also achieved almost the same RMSE with $k = 0.5$ within the 180 s time limit in four out of five trials. For the `Student` dataset, the CPA_{MC} method provided the best RMSE despite reaching the 180 s time limit in all five trials. On the other hand, our CPA_{2W} method with $k = 0.5$ obtained roughly the same RMSE, and completed the computation within the 180 s time limit in three out of five trials.

Table 5 gives results of logistic regression on the public `SPECT` and `Breast` datasets. For the `SPECT` dataset, the L1-Reg method was the fastest, but it violated the multicollinearity constraint and yielded the worst AUC. The CPA_{MC} method satisfied the multicollinearity constraint, and its AUC was better than that of the L1-Reg method; however, the CPA_{MC} method failed to complete the computation within the 180 s time limit in all five trials. In contrast, our CPA_{2W} method with all $k \in \{ 0.0, 0.5, 1.0, 2.0 \}$ was superior to the L1-Reg and CPA_{MC} methods in terms of RMSE. More importantly, our CPA_{2W} method with $k = 0.5$ attained the best AUC in only 0.32 s. For the `Breast` dataset, the L1-Reg method was the fastest but violated the multicollinearity constraint. While the CPA_{MC} method obtained the worst AUC, our CPA_{2W} method with $k \in \{ 0.0, 0.5, 1.0 \}$ provided better AUCs within 10 s than did the L1-Reg and CPA_{MC} methods.

6 Conclusion

We considered selecting the best subset of explanatory variables subject to the multicollinearity constraint for linear and logistic regression models. For this problem, we developed the two-way cutting-plane algorithm, which uses two types of cutting planes to approximate the nonlinear objective function and to remove collinear subsets of explanatory variables. We also proved that our algorithm outputs an ε-optimal solution within a finite number of iterations.

In computational experiments using synthetic and public datasets, our method consistently found solutions that satisfy the multicollinearity constraint, and the obtained solutions were frequently better in terms of predictive performance than those of the L_1-regularized estimation (Friedman et al., 2010) and the previous cutting-plane algorithm (Tamura et al., 2017) for eliminating multicollinearity. Moreover, increasing the weight of the L_2-regularization term in the objective function significantly accelerated our algorithm while maintaining high predictive performance.

The first advantage of our method is the acceleration of selecting the best subset of explanatory variables subject to the multicollinearity constraint. In contrast to many heuristic algorithms, our method can compute solutions with guaranteed global optimality, and its computation is faster than the previous cutting-plane algorithm (Tamura et al., 2017). The second advantage of our method is the improved accuracy of estimating regression coefficients by eliminating multicollinearity. This advantage is especially important when analyzing the regression results to make important decisions for practical purposes.

A future direction of study will be to further improve the computational efficiency of our cutting-plane algorithm. For example, when we need to add a huge number of linear constraints, it will be important to select and manage a portion of the active constraints. Another direction for future research is to extend our method to the estimation of other statistical and machine learning models, such as generalized linear models, support vector machines, and neural networks.

Acknowledgments

To be included after the peer review is completed.

References

Sahand Asgharieh Ahari and Burak Kocuk. A mixed-integer exponential cone programming formulation for feature subset selection in logistic regression. *EURO Journal on Computational Optimization*, 11:100069, 2023.

Lauren Berk and Dimitris Bertsimas. Certifiably optimal sparse principal component analysis. *Mathematical Programming Computation*, 11:381–420, 2019.

Dimitris Bertsimas and Angela King. OR forum—An algorithmic approach to linear regression. *Operations Research*, 64(1):2–16, 2016.

Dimitris Bertsimas and Angela King. Logistic regression: From art to science. *Statistical Science*, pp. 367–384, 2017.

Dimitris Bertsimas and Michael Lingzhi Li. Scalable holistic linear regression. *Operations Research Letters*, 48(3):203–208, 2020.

Dimitris Bertsimas, Angela King, and Rahul Mazumder. Best subset selection via a modern optimization lens. *The Annals of Statistics*, 44(2):813–852, 2016.

Dimitris Bertsimas, Jean Pauphilet, and Bart Van Parys. Sparse regression: Scalable algorithms and empirical performance. *Statistical Science*, 35(4):555–578, 2020.

Dimitris Bertsimas, Jean Pauphilet, and Bart Van Parys. Sparse classification: A scalable discrete optimization perspective. *Machine Learning*, 110:3177–3209, 2021.

Robert E Bixby. A brief history of linear and mixed-integer programming computation. *Documenta Mathematica*, 2012:107–121, 2012.

Stephen P Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, 2004.

Emilio Carrizosa, Alba V Olivares-Nadal, and Pepa Ramírez-Cobo. Integer constraints for enhancing interpretability in linear regression. *SORT—Statistics and Operations Research Transactions*, 44(1):67–98, 2020.

Emilio Carrizosa, Cristina Molero-Río, and Dolores Romero Morales. Mathematical optimization in classification and regression trees. *TOP*, 29(1):5–33, 2021.

Jireh Yi-Le Chan, Steven Mun Hong Leow, Khean Thye Bea, Wai Khuen Cheng, Seuk Wai Phoong, Zeng-Wei Hong, and Yen-Lin Chen. Mitigating the multicollinearity problem and its machine learning approach: A review. *Mathematics*, 10(8):1283, 2022.

Girish Chandrashekar and Ferat Sahin. A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28, 2014.

Samprit Chatterjee and Ali S Hadi. *Regression Analysis by Example*. John Wiley & Sons, New Jersey, 2015.

Seokhyun Chung, Young Woong Park, and Taesu Cheong. A mathematical programming approach for integrated multiple linear regression subset selection and validation. *Pattern Recognition*, 108:107565, 2020.

Alison Cozad, Nikolaos V Sahinidis, and David C Miller. Learning surrogate models for simulation-based optimization. *AIChE Journal*, 60(6):2211–2227, 2014.

Antoine Dedieu, Hussein Hazimeh, and Rahul Mazumder. Learning sparse classifiers: Continuous and mixed integer optimization perspectives. *Journal of Machine Learning Research*, 22(135):1–47, 2021.

Jerome H Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33:1–22, 2010.

Claudio Gambella, Bissan Ghaddar, and Joe Naoum-Sawaya. Optimization problems for machine learning: A survey. *European Journal of Operational Research*, 290(3):807–828, 2021.

Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

Trevor Hastie, Robert Tibshirani, and Ryan Tibshirani. Best subset, forward stepwise or lasso? Analysis and recommendations based on extensive comparisons. *Statistical Science*, 35(4):579–592, 2020.

Hussein Hazimeh and Rahul Mazumder. Fast best subset selection: Coordinate descent and local combinatorial optimization algorithms. *Operations Research*, 68(5):1517–1537, 2020.

Hussein Hazimeh, Rahul Mazumder, and Ali Saab. Sparse regression at scale: Branch-and-bound rooted in first-order optimization. *Mathematical Programming*, 196(1):347–388, 2022.

Ronald R Hocking. The analysis and selection of variables in linear regression. *Biometrics*, pp. 1–49, 1976.

Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning: with Applications in R*. Springer Publishing Company, Incorporated, New York, 2014. ISBN 1461471370.

Ian T Jolliffe. A note on the use of principal components in regression. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 31(3):300–303, 1982.

Ken Kobayashi, Yuichi Takano, and Kazuhide Nakata. Cardinality-constrained distributionally robust portfolio optimization. *European Journal of Operational Research*, 309(3):1173–1182, 2023.

Thorsten Koch, Timo Berthold, Jaap Pedersen, and Charlie Vanaret. Progress in mathematical programming solvers from 2001 to 2020. *EURO Journal on Computational Optimization*, 10:100031, 2022.

Hiroshi Konno and Rei Yamamoto. Choosing the best set of variables in regression analysis using integer programming. *Journal of Global Optimization*, 44:273–282, 2009.

Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P Trevino, Jiliang Tang, and Huan Liu. Feature selection: A data perspective. *ACM Computing Surveys (CSUR)*, 50(6):1–45, 2017.

Huan Liu and Hiroshi Motoda. *Computational methods of feature selection*. CRC Press, Florida, 2007.

Jiachang Liu, Sam Rosen, Chudi Zhong, and Cynthia Rudin. OKRidge: Scalable optimal k-sparse ridge regression. *Advances in Neural Information Processing Systems*, 36, 2024.

Sebastián Maldonado, Juan Pérez, Richard Weber, and Martine Labbé. Feature selection for support vector machines via mixed integer linear programming. *Information Sciences*, 279:163–175, 2014.

Kelly Markelle, Longjohn Rachel, and Nottingham Kolby. The uci machine learning repository, 2024. URL `https://archive.ics.uci.edu`.

Rahul Mazumder, Peter Radchenko, and Antoine Dedieu. Subset selection with shrinkage: Sparse linear modeling when the SNR is low. *Operations Research*, 71(1):129–147, 2023.

Ryuhei Miyashiro and Yuichi Takano. Mixed integer second-order cone programming formulations for variable selection in linear regression. *European Journal of Operational Research*, 247(3):721–731, 2015a.

Ryuhei Miyashiro and Yuichi Takano. Subset selection by Mallows' C_p: A mixed integer programming approach. *Expert Systems with Applications*, 42(1):325–331, 2015b.

Mizuho Naganuma, Yuichi Takano, and Ryuhei Miyashiro. Feature subset selection for ordered logit model via tangent-plane-based approximation. *IEICE Transactions on Information and Systems*, 102(5):1046–1053, 2019.

Young Woong Park and Diego Klabjan. Subset selection for multiple linear regression via optimization. *Journal of Global Optimization*, 77(3):543–574, 2020.

Hiroki Saishu, Kota Kudo, and Yuichi Takano. Sparse Poisson regression via mixed-integer optimization. *Plos One*, 16(4):e0249916, 2021.

Hiroki Saishu, Kota Kudo, and Yuichi Takano. Cutting-plane algorithm for estimation of sparse Cox proportional hazards models. *Top*, 32(1):57–82, 2024.

Toshiki Sato, Yuichi Takano, Ryuhei Miyashiro, and Akiko Yoshise. Feature subset selection for logistic regression via mixed integer optimization. *Computational Optimization and Applications*, 64:865–880, 2016.

Toshiki Sato, Yuichi Takano, and Ryuhei Miyashiro. Piecewise-linear approximation for feature subset selection in a sequential logit model. *Journal of the Operations Research Society of Japan*, 60(1):1–14, 2017.

Yuichi Takano and Ryuhei Miyashiro. Best subset selection via cross-validation criterion. *TOP*, 28(2):475–488, 2020.

Ryuta Tamura, Ken Kobayashi, Yuichi Takano, Ryuhei Miyashiro, Kazuhide Nakata, and Tomomi Matsui. Best subset selection for eliminating multicollinearity. *Journal of the Operations Research Society of Japan*, 60(3):321–336, 2017.

Ryuta Tamura, Ken Kobayashi, Yuichi Takano, Ryuhei Miyashiro, Kazuhide Nakata, and Tomomi Matsui. Mixed integer quadratic optimization formulations for eliminating multicollinearity based on variance inflation factor. *Journal of Global Optimization*, 73:431–446, 2019.

Ryuta Tamura, Yuichi Takano, and Ryuhei Miyashiro. Mixed-integer linear optimization formulations for feature subset selection in kernel SVM classification. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 107(8):1151–1162, 2024.

Berk Ustun and Cynthia Rudin. Supersparse linear integer models for optimized medical scoring systems. *Machine Learning*, 102:349–391, 2016.

Akihisa Watanabe, Ryuta Tamura, Yuichi Takano, and Ryuhei Miyashiro. Branch-and-bound algorithm for optimal sparse canonical correlation analysis. *Expert Systems with Applications*, 217:119530, 2023.

Svante Wold, Michael Sjöström, and Lennart Eriksson. PLS-regression: A basic tool of chemometrics. *Chemometrics and Intelligent Laboratory Systems*, 58(2):109–130, 2001.

Weijun Xie and Xinwei Deng. Scalable algorithms for the sparse ridge regression. *SIAM Journal on Optimization*, 30(4):3359–3386, 2020.

Mohammad Nasir Yarahmadi, Seyed Ali MirHassani, and Farnaz Hooshmand. Handling the significance of regression coefficients via optimization. *Expert Systems with Applications*, 238:121910, 2024.