Journal of the Operations Research Society of Japan
Vol. 60, No. 3, July 2017, pp. 321–336

© The Operations Research Society of Japan

BEST SUBSET SELECTION FOR ELIMINATING MULTICOLLINEARITY

Ryuta Tamura
Tokyo University of Agriculture and Technology

Ken Kobayashi
Fujitsu Laboratories Ltd.

Yuichi Takano
Senshu University

Ryuhei Miyashiro
Tokyo University of Agriculture and Technology

Kazuhide Nakata
Tokyo Institute of Technology

Tomomi Matsui
Tokyo Institute of Technology

(Received July 27, 2016; Revised December 2, 2016)

Abstract This paper proposes a method for eliminating multicollinearity from linear regression models. Specifically, we select the best subset of explanatory variables subject to the upper bound on the condition number of the correlation matrix of selected variables. We first develop a cutting plane algorithm that, to approximate the condition number constraint, iteratively appends valid inequalities to the mixed integer quadratic optimization problem. We also devise a mixed integer semidefinite optimization formulation for best subset selection under the condition number constraint. Computational results demonstrate that our cutting plane algorithm frequently provides solutions of better quality than those obtained using local search algorithms for subset selection. Additionally, subset selection by means of our optimization formulation succeeds when the number of candidate explanatory variables is small.

Keywords: Optimization, statistics, subset selection, multicollinearity, linear regression, mixed integer semidefinite optimization

1. Introduction

Multicollinearity, which exists when two or more explanatory variables in a regression model are highly correlated, is a frequently encountered problem in multiple regression analysis [11, 14, 24]. Such an interrelationship among explanatory variables obscures their relationship with the explained variable, leading to computational instability in model estimation. Moreover, the reliability of the regression analysis is decreased in the presence of multicollinearity by the low quality of the resultant estimates.

Several approaches can be used to avoid the deleterious effects of multicollinearity [7]. One approach is orthogonal transformation through procedures such as principal component regression [19, 25] and partial least squares regression [35, 36]. In this approach, a set of correlated variables is transformed into a set of linearly uncorrelated variables (i.e., principal components) for use in a regression model. Orthogonal transformation can enhance the computational stability of model estimation but often leads to worse predictive performance and results that are strongly influenced by the presence of outliers [12, 15].

Another approach is penalized regression, such as ridge regression [16], lasso [32], and elastic net [38]. This approach introduces a penalty function to shrink regression coefficient estimates toward zero. Penalized regression helps prevent regression models from overfitting noisy datasets and, accordingly, is effective for achieving high predictive performance. However, the penalty functions produce biased estimates, which are undesirable from the standpoint of model interpretation [5, 6].

This paper focuses on subset selection, which is a simple but effective approach for eliminating multicollinearity. Conventionally in this approach, explanatory variables are deleted iteratively through the use of indicators for detecting multicollinearity, such as condition number of the correlation matrix and variance inflation factor [7]. Chong and Jun [8] have compared the performance of subset selection methods with that of partial least squares regression and suggested that a goodness-of-fit measure for evaluating a subset regression model should be chosen carefully when multicollinearity is present. For subset selection in the presence of multicollinearity, several researchers [9, 18] have proposed goodness-of-fit measures based on ridge regression. It is notable, however, that commonly used subset selection methods are heuristic algorithms, such as stepwise regression [10]; hence, they do not necessarily find the best subset of variables in terms of a given goodness-of-fit measure.

The mixed integer optimization (MIO) approach to subset selection has recently received much attention due to advances in algorithms and hardware [5, 6, 20–22, 27–30]. In contrast to heuristic algorithms, the MIO approach has the potential to provide the best subset of variables under a given goodness-of-fit measure. To deal with multicollinearity, Bertsimas and King [5] use a cutting plane strategy, which iteratively adds constraints for deleting subsets of collinear variables. However, this strategy requires solving an enormous number of mixed integer quadratic optimization (MIQO) problems when multicollinearity exists in many different sets of variables.

The aim of this paper is to devise a more sophisticated MIO approach to best subset selection for eliminating multicollinearity. In particular, this paper addresses the following problem: Find a subset of variables that minimizes the residual sum of squares under the constraint that the condition number of the associated correlation matrix is bounded. Our first approach is to develop a high-performance cutting plane algorithm for subset selection. Our algorithm effectively uses a backward elimination method that searches a smaller subset of collinear variables to strengthen valid inequalities. On the other hand, the cutting plane algorithm must solve an exponential number of MIQO problems, which are NP-hard, in a worst-case situation. From this perspective, it makes a sense to pose the subset selection problem as a single MIO problem. In light of this fact, our second approach is to propose a mixed integer semidefinite optimization (MISDO) formulation for subset selection. In this approach, we must solve only a single MISDO problem, whereas the cutting plane algorithm must repeatedly solve a series of MIQO problems. Current MISDO algorithms do not deliver high computational performance, but their performance improvement is expected in the future. Furthermore, to increase computational efficiency, we consider incorporating constraints based on the normal equations into the MISDO problem.

The effectiveness of our MIO approaches is assessed through computational experiments using several datasets from the UCI Machine Learning Repository [23]. The computational results demonstrate that our cutting plane algorithm frequently gave a better subset of variables than did the conventional local search algorithms. In addition, we succeeded in solving some of our MISDO problems for subset selection when the number of candidate explanatory variables was less than 26.

2. Linear Regression and Multicollinearity

This section contains a brief review of linear regression and subset selection for eliminating multicollinearity.

2.1. Linear regression

Let us suppose that we are given n samples, $(\boldsymbol{x}_i, y_i)$ for $i = 1, 2, \ldots, n$. Here, $\boldsymbol{x}_i := (x_{i1}, x_{i2}, \ldots, x_{ip})^\top$ is a vector composed of p explanatory variables, and y_i is an explained variable for each sample $i = 1, 2, \ldots, n$.

We focus on the following linear regression model:

$$y_i = \boldsymbol{a}^\top \boldsymbol{x}_i + \varepsilon_i = \sum_{j=1}^{p} a_j x_{ij} + \varepsilon_i \quad (i = 1, 2, \ldots, n),$$

where $\boldsymbol{a} := (a_1, a_2, \ldots, a_p)^\top$ is a vector of regression coefficients to be estimated and ε_i is a prediction residual for each sample $i = 1, 2, \ldots, n$. We assume here that all explanatory and explained variables are standardized so that $(\sum_{i=1}^n x_{ij})/n = (\sum_{i=1}^n y_i)/n = 0$ and $(\sum_{i=1}^n (x_{ij})^2)/n = (\sum_{i=1}^n (y_i)^2)/n = 1$ for all $j = 1, 2, \ldots, p$. Therefore, no intercept (constant term) is present in the regression model.

The above linear regression model can be rewritten as

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{a} + \boldsymbol{\varepsilon},$$

where $\boldsymbol{y} := (y_1, y_2, \ldots, y_n)^\top$, $\boldsymbol{\varepsilon} := (\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n)^\top$, and

$$\boldsymbol{X} := \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix}.$$

To estimate the regression coefficients, $\boldsymbol{a}$, the ordinary least squares method minimizes the residual sum of squares (RSS):

$$\boldsymbol{\varepsilon}^\top \boldsymbol{\varepsilon} = (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{a})^\top (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{a}). \tag{2.1}$$

After partial differentiation, the ordinary least squares method is equivalent to solving a system of linear equations:

$$\boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{a} = \boldsymbol{X}^\top \boldsymbol{y}. \tag{2.2}$$

This is the well-known normal equation.

2.2. Subset selection for eliminating multicollinearity

We use the condition number of the correlation matrix $\boldsymbol{R} := (r_{j\ell};\ j, \ell = 1, 2 \ldots, p)$ to detect multicollinearity. Because the explanatory variables are standardized, the correlation matrix is calculated as

$$\boldsymbol{R} = \frac{\boldsymbol{X}^\top \boldsymbol{X}}{n},$$

and its condition number is defined as

$$\mathrm{cond}(\boldsymbol{R}) := \begin{cases} \dfrac{\lambda_{\max}(\boldsymbol{R})}{\lambda_{\min}(\boldsymbol{R})} & (\lambda_{\min}(\boldsymbol{R}) > 0), \\ \\ +\infty & (\lambda_{\min}(\boldsymbol{R}) = 0), \end{cases} \tag{2.3}$$

where $\lambda_{\min}(\boldsymbol{R})$ and $\lambda_{\max}(\boldsymbol{R})$ are the minimum and maximum eigenvalues of matrix $\boldsymbol{R}$, respectively. It follows from $\mathrm{cond}(\boldsymbol{R}) = \mathrm{cond}(\boldsymbol{X}^\top \boldsymbol{X})$ that when $\mathrm{cond}(\boldsymbol{R})$ is very large, the coefficient matrix of equations (2.2) is ill-conditioned, which implies that the regression coefficient estimates are subject to large numerical errors.

To compute accurate estimates, we consider selecting a subset $S \subseteq \{1, 2, \ldots, p\}$ of candidate explanatory variables. Let us denote by $\boldsymbol{R}_S$ the correlation sub-matrix of a subset of variables, that is, $\boldsymbol{R}_S := (r_{j\ell};\ j, \ell \in S)$. To avoid multicollinearity, the condition number of the sub-matrix should not exceed a user-defined parameter κ (> 1). The subset selection problem determines a subset S of explanatory variables so that the residual sum of squares of a subset regression model is minimized:

$$\begin{aligned} \underset{\boldsymbol{a},\, S}{\text{minimize}} \quad & \sum_{i=1}^{n} \left(y_i - \sum_{j \in S} a_j x_{ij} \right)^2 && (2.4) \\ \text{subject to} \quad & \mathrm{cond}(\boldsymbol{R}_S) \le \kappa, && (2.5) \\ & S \subseteq \{1, 2, \ldots, p\}. && (2.6) \end{aligned}$$

3. Cutting Plane Algorithm

The subset selection problem (2.4)–(2.6) can be converted into an MIO problem. Let $\boldsymbol{z} := (z_1, z_2, \ldots, z_p)^\top$ be a vector of 0-1 decision variables for selecting explanatory variables; that is, $z_j = 1$ if $j \in S$; otherwise, $z_j = 0$. The correlation sub-matrix is then written as $\boldsymbol{R}(\boldsymbol{z}) := (r_{j\ell};\ z_j = z_\ell = 1)$. Consequently, the subset selection problem is formulated as an MIO problem,

$$\begin{aligned} \underset{\boldsymbol{a},\, \boldsymbol{z}}{\text{minimize}} \quad & (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{a})^\top (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{a}) && (3.1) \\ \text{subject to} \quad & z_j = 0 \ \Rightarrow \ a_j = 0 \quad (j = 1, 2, \ldots, p), && (3.2) \\ & \mathrm{cond}(\boldsymbol{R}(\boldsymbol{z})) \le \kappa, && (3.3) \\ & \boldsymbol{z} \in \{0, 1\}^p. && (3.4) \end{aligned}$$

Here, if $z_j = 0$, then the jth explanatory variable is deleted from the regression model because its coefficient is set to zero by the logical implications (3.2). It is known that these logical implications can be represented by using a big-M method or a special ordered set type 1 (SOS1) constraint [3, 4].

A cutting plane algorithm first removes the condition number constraint (3.3) from the problem. Hence its feasible set is expressed as

$$\mathcal{F}_0 := \{(\boldsymbol{a}, \boldsymbol{z}) \mid \text{constraints (3.2) and (3.4)}\}, \qquad (3.5)$$

and the problem reduces to an MIQO problem, which can be handled by using standard optimization software. The basic strategy of the algorithm involves repeatedly solving such relaxed MIQO problems and iteratively adding valid inequalities, instead of imposing the condition number constraint (3.3), to the MIQO problems.

Our cutting plane algorithm first checks whether $\mathrm{cond}(\boldsymbol{R}) \le \kappa$. If $\mathrm{cond}(\boldsymbol{R}) > \kappa$, the feasible set is updated to avoid selecting all candidate variables,

$$\mathcal{F}_1 := \mathcal{F}_0 \cap \{(\boldsymbol{a}, \boldsymbol{z}) \mid \boldsymbol{1}^\top \boldsymbol{z} \le p - 1\},$$

where $\boldsymbol{1} := (1, 1, \ldots, 1)^\top \in \mathbb{R}^p$.

Next we set $k \leftarrow 1$ and solve the following MIQO problem:

$$\underset{\boldsymbol{a},\boldsymbol{z}}{\text{minimize}} \quad (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{a})^\top(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{a}) \tag{3.6}$$

$$\text{subject to} \quad (\boldsymbol{a}, \boldsymbol{z}) \in \mathcal{F}_k, \tag{3.7}$$

where $\mathcal{F}_k$ is the feasible set, which will be updated based on equation (3.8), at the kth iteration. Let $(\boldsymbol{a}_1, \boldsymbol{z}_1)$ be an optimal solution to problem (3.6) and (3.7) with $k = 1$. We then set $k \leftarrow 2$. If $\mathrm{cond}(\boldsymbol{R}(\boldsymbol{z}_1)) > \kappa$, then the solution $\boldsymbol{z}_1$ can be separated from the feasible set as follows:

$$\mathcal{F}_k := \mathcal{F}_{k-1} \cap \{(\boldsymbol{a}, \boldsymbol{z}) \mid \boldsymbol{z}_{k-1}^\top \boldsymbol{z} \le \boldsymbol{z}_{k-1}^\top \boldsymbol{z}_{k-1} - 1\}. \tag{3.8}$$

After that, we solve the MIQO problem (3.6) and (3.7) again. This process is repeated by setting $k \leftarrow k + 1$ until we obtain a solution that satisfies the condition number constraint (3.3).

We next show that the condition number is not increased by deleting explanatory variables.

Lemma 3.1. *Suppose that* $\boldsymbol{z}, \bar{\boldsymbol{z}} \in \{0,1\}^p$. *If* $\boldsymbol{z} \ge \bar{\boldsymbol{z}}$, *then* $\mathrm{cond}(\boldsymbol{R}(\boldsymbol{z})) \ge \mathrm{cond}(\boldsymbol{R}(\bar{\boldsymbol{z}}))$.

Proof. It follows from the Cauchy's interlace theorem (see, e.g., Theorem 4.3.17 [17]) that

$$0 \le \lambda_{\min}(\boldsymbol{R}(\boldsymbol{z})) \le \lambda_{\min}(\boldsymbol{R}(\bar{\boldsymbol{z}})) \le \lambda_{\max}(\boldsymbol{R}(\bar{\boldsymbol{z}})) \le \lambda_{\max}(\boldsymbol{R}(\boldsymbol{z})),$$

which completes the proof. □

The next theorem states that the feasible set of the original problem (3.1)–(3.4) is contained in $\mathcal{F}_k$ for all k.

Theorem 3.1. *Suppose that* $\bar{\boldsymbol{z}} \in \{0,1\}^p$ *and* $\mathrm{cond}(\boldsymbol{R}(\bar{\boldsymbol{z}})) > \kappa$. *If* $\boldsymbol{z} \in \{0,1\}^p$ *satisfies* $\mathrm{cond}(\boldsymbol{R}(\boldsymbol{z})) \le \kappa$, *then it also satisfies* $\bar{\boldsymbol{z}}^\top \boldsymbol{z} \le \bar{\boldsymbol{z}}^\top \bar{\boldsymbol{z}} - 1$.

Proof. We prove the proposition by contradiction:

$$\begin{aligned}
\bar{\boldsymbol{z}}^\top \boldsymbol{z} > \bar{\boldsymbol{z}}^\top \bar{\boldsymbol{z}} - 1 \;&\Rightarrow\; \bar{\boldsymbol{z}}^\top(\boldsymbol{z} - \bar{\boldsymbol{z}}) \ge 0 \quad \because \boldsymbol{z}, \bar{\boldsymbol{z}} \in \{0,1\}^p \\
&\Rightarrow\; \boldsymbol{z} \ge \bar{\boldsymbol{z}} \\
&\Rightarrow\; \mathrm{cond}(\boldsymbol{R}(\boldsymbol{z})) \ge \mathrm{cond}(\boldsymbol{R}(\bar{\boldsymbol{z}})) \quad \because \text{Lemma 3.1} \\
&\Rightarrow\; \mathrm{cond}(\boldsymbol{R}(\boldsymbol{z})) > \kappa.
\end{aligned}$$

□

This cutting plane algorithm is developed based on Bertsimas and King [5] and requires solving a large number of MIQO problems. To reduce the number of MIQO problems to be solved, we develop stronger valid inequalities for approximating the condition number constraint (3.3). To this end, we employ a backward elimination method that searches a smaller subset of collinear variables. Specifically, it starts with an incumbent solution (e.g., $\boldsymbol{z}_{k-1}$) and deletes explanatory variables one by one on the basis of the RSS (2.1). Finally, we obtain $\bar{\boldsymbol{z}}$ ($\le \boldsymbol{z}_{k-1}$) such that $\mathrm{cond}(\boldsymbol{R}(\bar{\boldsymbol{z}})) > \kappa$. The feasible set is then updated:

$$\mathcal{F}_k := \mathcal{F}_{k-1} \cap \{(\boldsymbol{a}, \boldsymbol{z}) \mid \bar{\boldsymbol{z}}^\top \boldsymbol{z} \le \bar{\boldsymbol{z}}^\top \bar{\boldsymbol{z}} - 1\}. \tag{3.9}$$

This valid inequality cuts off all $\boldsymbol{z} \in \{0,1\}^p$ satisfying $\boldsymbol{z} \ge \bar{\boldsymbol{z}}$; therefore, it is stronger than the previous one (3.8) because $\boldsymbol{z}_{k-1} \ge \bar{\boldsymbol{z}}$.

Our cutting plane algorithm is summarized as follows:

Cutting plane algorithm for solving problem (3.1)–(3.4)

Step 0 (Initialization) Set $\boldsymbol{z}_0 := \boldsymbol{1}$ and $k \leftarrow 1$. Let $\mathcal{F}_0$ be a feasible set (3.5).

Step 1 (Multicollinearity detection) If $\mathrm{cond}(\boldsymbol{R}(\boldsymbol{z}_{k-1})) \le \kappa$, terminate the algorithm with the solution $\boldsymbol{z}_{k-1}$.

Step 2 (Backward elimination) Find a solution $\bar{\boldsymbol{z}} \in \{0,1\}^p$ such that $\bar{\boldsymbol{z}} \le \boldsymbol{z}_{k-1}$ and $\mathrm{cond}(\boldsymbol{R}(\bar{\boldsymbol{z}})) > \kappa$ by using a backward elimination method starting with $\boldsymbol{z}_{k-1}$.

Step 3 (Cut generation) Add cut (3.9) to update the feasible set $\mathcal{F}_k$.

Step 4 (Relaxed MIQO problem) Solve problem (3.6) and (3.7). Let $(\boldsymbol{a}_k, \boldsymbol{z}_k)$ be an optimal solution. Set $k \leftarrow k+1$ and return to Step 1.

We next prove the finite convergence of the algorithm.

Theorem 3.2. *Our cutting plane algorithm provides an optimal solution to problem* (3.1)–(3.4) *in a finite number of iterations.*

Proof. Step 3 removes $\boldsymbol{z}_{k-1}$ from $\mathcal{F}_{k-1}$ in each iteration; therefore, the algorithm terminates with a feasible solution (e.g., $\boldsymbol{z} = (1,0,0,\ldots,0)^\top$) after at most $2^p - p$ iterations.

Let $(\boldsymbol{a}^*, \boldsymbol{z}^*)$ be an optimal solution to problem (3.1)–(3.4). Theorem 3.1 guarantees that the feasible set of problem (3.1)–(3.4) is contained in $\mathcal{F}_k$ and hence that $(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{a}_k)^\top(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{a}_k) \le (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{a}^*)^\top(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{a}^*)$ for all k. Therefore, the algorithm provides an optimal solution to problem (3.1)–(3.4). □

4. Mixed Integer Semidefinite Optimization Approach

This section presents an MISDO approach to best subset selection for eliminating multicollinearity. The motivation behind this approach is to reduce the subset selection problem to a single MIO problem. As a result, we must handle only a single MISDO problem to find the best subset of explanatory variables. By contrast, the cutting plane algorithm, in a worst-case situation, must solve an exponential number of MIQO problems, which are NP-hard.

4.1. Formulation

A convex quadratic objective function (3.1) is expressed as a linear objective function with a positive semidefinite constraint [33, 34]. We begin by computing a decomposition of the form:

$$\boldsymbol{X}^\top \boldsymbol{X} = n\boldsymbol{R} = \boldsymbol{V}\boldsymbol{V}^\top,$$

where the square matrix $\boldsymbol{V} \in \mathbb{R}^{p \times p}$ can be created, e.g., by the Cholesky/eigenvalue decomposition. Introducing a scalar decision variable f to be minimized, we rewrite the associated constraint as a positive semidefinite constraint as follows:

$$(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{a})^\top(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{a}) \le f \iff \begin{pmatrix} \boldsymbol{I}_p & \boldsymbol{V}^\top \boldsymbol{a} \\ \boldsymbol{a}^\top \boldsymbol{V} & 2\boldsymbol{y}^\top \boldsymbol{X}\boldsymbol{a} - \boldsymbol{y}^\top \boldsymbol{y} + f \end{pmatrix} \succeq \boldsymbol{O},$$

where $\boldsymbol{I}_p$ is the identity matrix of size p, $\boldsymbol{O}$ is a zero matrix of appropriate size, and $\boldsymbol{A} \succeq \boldsymbol{B}$ means that $\boldsymbol{A} - \boldsymbol{B}$ is a positive semidefinite matrix.

We denote by $\mathrm{Diag}(\boldsymbol{x})$ the diagonal matrix whose diagonal entries are components of vector $\boldsymbol{x}$. We also denote by $\boldsymbol{A} \circ \boldsymbol{B}$ the Hadamard product of matrices $\boldsymbol{A}$ and $\boldsymbol{B}$. The next theorem shows that the condition number constraint (3.3) is expressed as positive semidefinite constraints based on the following matrices:

$$\mathrm{Diag}(\mathbf{1}-\boldsymbol{z}) = \begin{pmatrix} 1-z_1 & 0 & \cdots & 0 \\ 0 & 1-z_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1-z_p \end{pmatrix},$$

$$\boldsymbol{R} \circ \boldsymbol{z}\boldsymbol{z}^\top = \begin{pmatrix} r_{11}z_1z_1 & r_{12}z_1z_2 & \cdots & r_{1p}z_1z_p \\ r_{21}z_2z_1 & r_{22}z_2z_2 & \cdots & r_{2p}z_2z_p \\ \vdots & \vdots & \ddots & \vdots \\ r_{p1}z_pz_1 & r_{p2}z_pz_2 & \cdots & r_{pp}z_pz_p \end{pmatrix}.$$

Theorem 4.1. *Suppose that $\boldsymbol{z} \in \{0,1\}^p$. Then, $\mathrm{cond}(\boldsymbol{R}(\boldsymbol{z})) \le \kappa$ holds if and only if there exists $\lambda \in [1/\kappa, 1]$ such that*

$$\lambda \boldsymbol{I}_p - \mathrm{Diag}(\mathbf{1}-\boldsymbol{z}) \preceq \boldsymbol{R} \circ \boldsymbol{z}\boldsymbol{z}^\top \preceq \kappa\lambda \boldsymbol{I}_p. \tag{4.1}$$

Proof. Let q be the number of nonzero elements of $\boldsymbol{z}$. Without loss of generality, we assume that

$$\begin{cases} z_j = 1 & \text{for } j = 1,2,\ldots,q, \\ z_j = 0 & \text{for } j = q+1, q+2, \ldots, p. \end{cases} \tag{4.2}$$

Since $\boldsymbol{R}(\boldsymbol{z})$ is a positive semidefinite matrix whose diagonal entries are all one, it holds that $0 \le \lambda_{\min}(\boldsymbol{R}(\boldsymbol{z})) \le 1 \le \lambda_{\max}(\boldsymbol{R}(\boldsymbol{z}))$. It then follows from the definition (2.3) that $\mathrm{cond}(\boldsymbol{R}(\boldsymbol{z})) \le \kappa$ is equivalent to

$$\lambda_{\max}(\boldsymbol{R}(\boldsymbol{z})) \le \kappa \lambda_{\min}(\boldsymbol{R}(\boldsymbol{z})).$$

This also implies that $1/\kappa \le \lambda_{\max}(\boldsymbol{R}(\boldsymbol{z}))/\kappa \le \lambda_{\min}(\boldsymbol{R}(\boldsymbol{z}))$. Therefore, it is necessary to consider only $\lambda_{\min}(\boldsymbol{R}(\boldsymbol{z})) \in [1/\kappa, 1]$.

Using a positive semidefinite constraint for minimizing the maximal eigenvalue [33, 34], the condition number constraint can be converted as follows:

$$\begin{aligned} & \lambda_{\max}(\boldsymbol{R}(\boldsymbol{z})) \le \kappa\lambda_{\min}(\boldsymbol{R}(\boldsymbol{z})) \\ \iff & \exists \lambda \in [1/\kappa, 1],\ \lambda \le \lambda_{\min}(\boldsymbol{R}(\boldsymbol{z})) \text{ and } \lambda_{\max}(\boldsymbol{R}(\boldsymbol{z})) \le \kappa\lambda \\ \iff & \exists \lambda \in [1/\kappa, 1],\ \lambda \boldsymbol{I}_q \preceq \boldsymbol{R}(\boldsymbol{z}) \preceq \kappa\lambda \boldsymbol{I}_q \\ \iff & \exists \lambda \in [1/\kappa, 1],\ \begin{pmatrix} \lambda \boldsymbol{I}_q & \boldsymbol{O} \\ \boldsymbol{O} & (\lambda - 1)\boldsymbol{I}_{p-q} \end{pmatrix} \preceq \begin{pmatrix} \boldsymbol{R}(\boldsymbol{z}) & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} \end{pmatrix} \preceq \kappa\lambda \begin{pmatrix} \boldsymbol{I}_q & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{I}_{p-q} \end{pmatrix} \\ \iff & \exists \lambda \in [1/\kappa, 1],\ \lambda \boldsymbol{I}_p - \mathrm{Diag}(\mathbf{1}-\boldsymbol{z}) \preceq \boldsymbol{R} \circ \boldsymbol{z}\boldsymbol{z}^\top \preceq \kappa\lambda \boldsymbol{I}_p. \end{aligned}$$

□

To linearize the bilinear term $\boldsymbol{z}\boldsymbol{z}^\top$ in constraint (4.1), we introduce a symmetric matrix of decision variables:

$$\boldsymbol{W} = \begin{pmatrix} w_{11} & w_{21} & \cdots & w_{p1} \\ w_{21} & w_{22} & \cdots & w_{p2} \\ \vdots & \vdots & \ddots & \vdots \\ w_{p1} & w_{p2} & \cdots & w_{pp} \end{pmatrix}.$$

It is known that when $\boldsymbol{z} \in \{0,1\}^p$, $w_{j\ell} = z_j z_\ell$ can be rewritten by means of its convex and concave envelopes as follows [2, 26]:

$$w_{j\ell} \geq 0,\ w_{j\ell} \geq z_j + z_\ell - 1,\ w_{j\ell} \leq z_j,\ w_{j\ell} \leq z_\ell.$$

Consequently, the subset selection problem is cast into an MISDO problem,

$$\begin{aligned}
&\underset{\boldsymbol{a}, f, \lambda, \boldsymbol{W}, \boldsymbol{z}}{\text{minimize}} && f && (4.3)\\
&\text{subject to} && \begin{pmatrix} \boldsymbol{I}_p & \boldsymbol{V}^\top \boldsymbol{a} \\ \boldsymbol{a}^\top \boldsymbol{V} & 2\boldsymbol{y}^\top \boldsymbol{X}\boldsymbol{a} - \boldsymbol{y}^\top \boldsymbol{y} + f \end{pmatrix} \succeq \boldsymbol{O}, && (4.4)\\
& && z_j = 0 \ \Rightarrow \ a_j = 0 \quad (j = 1, 2, \ldots, p), && (4.5)\\
& && \lambda \boldsymbol{I}_p - \mathrm{Diag}(\boldsymbol{1} - \boldsymbol{z}) \preceq \boldsymbol{R} \circ \boldsymbol{W} \preceq \kappa\lambda \boldsymbol{I}_p, && (4.6)\\
& && w_{jj} = z_j \quad (j = 1, 2, \ldots, p), && (4.7)\\
& && w_{j\ell} \geq 0,\ w_{j\ell} \geq z_j + z_\ell - 1,\ w_{j\ell} \leq z_j,\ w_{j\ell} \leq z_\ell && \\
& && \qquad\qquad (j, \ell = 1, 2, \ldots, p;\ j > \ell), && (4.8)\\
& && 1/\kappa \leq \lambda \leq 1, \quad \boldsymbol{z} \in \{0,1\}^p. && (4.9)
\end{aligned}$$

4.2. Normal-equation-based constraints

MISDO problems can be handled by a branch-and-bound procedure, but it involves solving a large number of relaxed semidefinite optimization problems. To improve computational efficiency, we consider including the normal equations (2.2) as the constraints of the MISDO problem. More precisely, when the jth explanatory variable is selected, the jth normal equation is placed as follows:

$$\begin{aligned}
&\boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{a} + \boldsymbol{s} = \boldsymbol{X}^\top \boldsymbol{y}, && (4.10)\\
&z_j = 1 \ \Rightarrow \ s_j = 0 \quad (j = 1, 2, \ldots, p), && (4.11)
\end{aligned}$$

where $\boldsymbol{s} = (s_1, s_2, \ldots, s_p)^\top$ is a vector of auxiliary decision variables.

The next theorem shows that constraints (4.10) and (4.11) are necessary optimality conditions for problem (3.1)–(3.4).

Theorem 4.2. *Let* $(\boldsymbol{a}^*, \boldsymbol{z}^*)$ *be an optimal solution to problem* (3.1)–(3.4). *There exists* $\boldsymbol{s}^* = (s_1^*, s_2^*, \ldots, s_p^*)^\top$ *such that*

$$\begin{aligned}
&\boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{a}^* + \boldsymbol{s}^* = \boldsymbol{X}^\top \boldsymbol{y},\\
&z_j^* = 1 \ \Rightarrow \ s_j^* = 0 \quad (j = 1, 2, \ldots, p).
\end{aligned}$$

Proof. Without loss of generality, we can assume that

$$\boldsymbol{z}^* = \begin{pmatrix} \boldsymbol{1} \\ \boldsymbol{0} \end{pmatrix}, \ \boldsymbol{1} \in \mathbb{R}^q, \ \boldsymbol{0} \in \mathbb{R}^{p-q}.$$

According to $\boldsymbol{z}^*$, we partition $\boldsymbol{X}$ and $\boldsymbol{a}^*$ as follows:

$$\begin{aligned} \boldsymbol{X} &= \begin{pmatrix} \boldsymbol{X}_1 & \boldsymbol{X}_2 \end{pmatrix}, \ \boldsymbol{X}_1 \in \mathbb{R}^{n \times q}, \ \boldsymbol{X}_2 \in \mathbb{R}^{n \times (p-q)}, \\ \boldsymbol{a}^* &= \begin{pmatrix} \boldsymbol{a}_1^* \\ \boldsymbol{a}_2^* \end{pmatrix}, \ \boldsymbol{a}_1^* \in \mathbb{R}^q, \ \boldsymbol{a}_2^* \in \mathbb{R}^{p-q}. \end{aligned}$$

Because $(\boldsymbol{a}^*, \boldsymbol{z}^*)$ is an optimal solution to problem (3.1)–(3.4), we have $\boldsymbol{a}_2^* = \boldsymbol{0}$. Moreover, $\boldsymbol{a}_1^*$ minimizes RSS (2.1); that is, the following holds for $\boldsymbol{a}_1^*$ and the associated normal equation:

$$\boldsymbol{X}_1^\top \boldsymbol{X}_1 \boldsymbol{a}_1^* = \boldsymbol{X}_1^\top \boldsymbol{y}.$$

Now we define $\boldsymbol{s}^*$ as follows:

$$\boldsymbol{s}^* = \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{X}_2^\top \boldsymbol{y} - \boldsymbol{X}_2^\top \boldsymbol{X}_1 \boldsymbol{a}_1^* \end{pmatrix}.$$

It then follows that

$$\begin{aligned} &\boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{a}^* + \boldsymbol{s}^* = \begin{pmatrix} \boldsymbol{X}_1^\top \boldsymbol{X}_1 \boldsymbol{a}_1^* \\ \boldsymbol{X}_2^\top \boldsymbol{X}_1 \boldsymbol{a}_1^* \end{pmatrix} + \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{X}_2^\top \boldsymbol{y} - \boldsymbol{X}_2^\top \boldsymbol{X}_1 \boldsymbol{a}_1^* \end{pmatrix} = \begin{pmatrix} \boldsymbol{X}_1^\top \boldsymbol{y} \\ \boldsymbol{X}_2^\top \boldsymbol{y} \end{pmatrix} = \boldsymbol{X}^\top \boldsymbol{y}, \\ &z_j^* = 1 \ \Rightarrow \ s_j^* = 0 \quad (j = 1, 2, \ldots, p). \end{aligned}$$

□

5. Computational Results

In this section, we assess the computational performance of our mixed integer optimization approaches to best subset selection for eliminating multicollinearity.

We downloaded six datasets for regression analysis from the UCI Machine Learning Repository [23]. Tables 1 lists the instances used for computational experiments, where n and p are the number of samples and number of candidate explanatory variables, respectively. In the `SolarFlareC` instance, C-class flares production was employed as an explained variable. In the `ForestFires` instance, interaction terms were created from the variables of the x-axis and y-axis spatial coordinates. Each categorical variable was transformed into one or more dummy variables. All explanatory and explained variables were standardized to a mean of zero and standard deviation of one as mentioned in Section 2.1. Samples containing missing values and redundant variables having the same value in all samples were eliminated.

5.1. Computational performance of cutting plane algorithms

We first compare the computational performance of the cutting plane algorithms with that of conventional local search algorithms for subset selection. The algorithms used in the comparison are listed below:

FwS Forward selection method: Starts with $S = \emptyset$ and iteratively adds the variable j (i.e., $S \leftarrow S \cup \{j\}$) that leads to the largest decrease in RSS (2.1); this operation is repeated while $\mathrm{cond}(\boldsymbol{R}_S) \leq \kappa$ is satisfied.

Table 1: List of instances

Abbreviation	n	p	Original dataset [23]
Servo	167	19	Servo
AutoMPG	392	25	Auto MPG
SolarFlareC	1066	26	Solar Flare (C-class flares production)
BreastCancer	194	32	Breast Cancer Wisconsin
ForestFires	517	63	Forest Fires
Automobile	159	65	Automobile

BwE Backward elimination method: Starts with $S = \{1, 2, \dots, p\}$ and iteratively eliminates the variable j (i.e., $S \leftarrow S \setminus \{j\}$) that leads to the smallest increase in RSS (2.1); this operation is repeated until $\mathrm{cond}(\boldsymbol{R}_S) \leq \kappa$ holds.

CPA Cutting plane algorithm that omits Step 2 (Backward elimination) and appends cut (3.8).

CPA$_{\mathbf{BwE}}$ Cutting plane algorithm that appends cut (3.9) strengthened by means of the backward elimination method.

These computations were performed on a Linux computer with an Intel Xeon W3520 CPU (2.66 GHz) and 12 GB memory. The algorithms FwS and BwE were implemented in MATLAB R2013a (http://www.mathworks.com/products/matlab/); the algorithms CPA and CPA$_{\mathrm{BwE}}$ were implemented in Python 2.7.3, with Gurobi Optimizer 5.6.0 (http://www.gurobi.com) used to solve relaxed MIQO problems (3.6) and (3.7). Here the logical implications (3.2) were incorporated in the form of SOS1 constraints, which imply that at most one element in the set can have a nonzero value. Specifically, the SOS1 constraint is imposed on $\{1 - z_j, a_j\}$ $(j = 1, 2, \dots, p)$ with the SOS type 1 function implemented in Gurobi Optimizer. Therefore, if $z_j = 0$, then $1 - z_j$ has a nonzero value and a_j must be zero from the SOS1 constraints. Chatterjee and Hadi [7] mention that when the value of the condition number exceeds 225, the deleterious effects of multicollinearity in the data become strong. Hence, the upper bound on the condition number was set as $\kappa = 100$ or 225.

Tables 2 and 3 show the computational results obtained using the four algorithms with $\kappa = 100$ and 225, respectively. The column labeled "R^2" shows the value of the coefficient of determination of a subset regression model built by each method, i.e.,

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \sum_{j \in S} a_j x_{ij})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2},$$

where $\bar{y} := (\sum_{i=1}^{n} y_i)/n$. Note here that the largest R^2 values for each instance are indicated in bold. The column labeled "Cond" shows the condition number of the correlation submatrix $\boldsymbol{R}_S$, and the column labeled "$|S|$" shows the number of selected explanatory variables. The column labeled "#Iter" shows the number of iterations in the cutting plane algorithms. The column labeled "Time (s)" shows the computation time in seconds. Note that the computation of the cutting plane algorithm was terminated if it did not finish by itself within 10000 s. In such cases, the best feasible solution obtained within 10000 s was used as the result and "N/A" means that no feasible solution was found.

We can see from Tables 2 and 3 that the local search algorithms FwS and BwE finished their computations within 2 s for all the instances. Their obtained solutions were, however, frequently inferior to those of the cutting plane algorithms, especially when $\kappa = 100$ (see Table 2); for the BreastCancer instance, CPA$_{\mathrm{BwE}}$ gave an R^2 value of 0.29040, whereas FwS and BwE gave R^2 values of 0.27580 and 0.26572, respectively.

Table 2: Results of local search algorithms and cutting plane algorithms ($\kappa = 100$)

Instance	n	p	Method	R^2	Cond	$\lvert S \rvert$	#Iter	Time (s)
Servo	167	19	FwS	0.75862	63.2	14	—	0.09
			BwE	**0.75877**	39.0	15	—	0.04
			CPA	**0.75877**	38.4	15	76	0.93
			CPA_{BwE}	**0.75877**	39.9	15	16	0.42
AutoMPG	392	25	FwS	0.87335	87.6	21	—	0.17
			BwE	0.87429	86.2	19	—	0.09
			CPA	**0.87430**	93.2	21	1284	325.20
			CPA_{BwE}	**0.87430**	92.8	21	84	14.11
SolarFlareC	1066	26	FwS	0.19713	4.3	19	—	0.23
			BwE	0.19705	2.0	15	—	0.87
			CPA	**0.19715**	18.7	19	4209	1246.97
			CPA_{BwE}	**0.19715**	34.3	19	331	84.59
BreastCancer	194	32	FwS	0.27580	91.8	15	—	0.17
			BwE	0.26572	61.8	9	—	0.27
			CPA	N/A			>4364	>10000.00
			CPA_{BwE}	**0.29040**	95.8	14	>1044	>10000.00
ForestFires	517	63	FwS	0.16399	99.9	52	—	1.68
			BwE	0.16481	53.1	50	—	1.32
			CPA	N/A			>5901	>10000.00
			CPA_{BwE}	**0.16537**	96.5	59	53	301.07
Automobile	159	65	FwS	0.96447	99.9	27	—	0.76
			BwE	0.96571	67.3	24	—	1.89
			CPA	N/A			>6960	>10000.00
			CPA_{BwE}	**0.96908**	72.9	25	>277	>10000.00

We can also see that the computation finished much faster for CPA_{BwE} than for CPA. The main reason for this is that the number of iterations required for CPA_{BwE} was significantly reduced by the strengthened cut (3.9). Indeed, in the case of `SolarFlareC` in Table 2, CPA arrived at an optimal solution in 1246.97 s after 4209 iterations, whereas CPA_{BwE} terminated with an optimal solution in 84.59 s after 331 iterations.

Note that when the computation is terminated due to the time limit of 10000 s, CPA does not provide a feasible solution because it is found only at the end of the algorithm. In contrast, CPA_{BwE} can find a feasible solution of good quality in the early stage of the algorithm by means of the backward elimination method. For this reason, CPA_{BwE} always provided the best solution among the four methods for all the instances in Tables 2 and 3.

5.2. Computational performance of MISDO approaches

Next we evaluate the computational performance of the following MISDO approaches:

MISDO MISDO formulation (4.3)–(4.9);

MISDO_{NE} MISDO formulation (4.3)–(4.9) with the normal-equation-based constraints (4.10) and (4.11).

Table 3: Results of local search algorithms and cutting plane algorithms ($\kappa = 225$)

Instance	n	p	Method	R^2	Cond	$\lvert S \rvert$	#Iter	Time (s)
`Servo`	167	19	FwS	**0.75877**	146.3	15	—	0.11
			BwE	**0.75877**	39.0	15	—	0.04
			CPA	**0.75877**	102.5	15	56	0.62
			CPA$_{\text{BwE}}$	**0.75877**	39.9	15	15	0.40
`AutoMPG`	392	25	FwS	**0.87438**	185.3	22	—	0.18
			BwE	**0.87438**	181.1	22	—	0.05
			CPA	**0.87438**	157.1	22	60	2.04
			CPA$_{\text{BwE}}$	**0.87438**	173.2	22	18	1.76
`SolarFlareC`	1066	26	FwS	0.19713	4.3	19	—	0.22
			BwE	0.19705	2.0	15	—	0.48
			CPA	**0.19715**	18.7	19	4209	1244.26
			CPA$_{\text{BwE}}$	**0.19715**	169.4	19	750	189.48
`BreastCancer`	194	32	FwS	0.30010	217.9	19	—	0.21
			BwE	0.26572	61.8	9	—	0.27
			CPA	N/A			>4369	>10000.00
			CPA$_{\text{BwE}}$	**0.30513**	215.6	20	287	288.59
`ForestFires`	517	63	FwS	**0.16556**	214.2	60	—	1.74
			BwE	**0.16556**	212.6	60	—	0.42
			CPA	**0.16556**	209.5	60	59	7.60
			CPA$_{\text{BwE}}$	**0.16556**	209.0	60	12	27.37
`Automobile`	159	65	FwS	0.97124	224.4	39	—	1.14
			BwE	0.97153	183.5	29	—	1.85
			CPA	N/A			>6973	>10000.00
			CPA$_{\text{BwE}}$	**0.97391**	224.4	36	>960	>10000.00

Here the logical implications (4.11) were represented by means of the big-M method,

$$-M(1-z_j) \leq s_j \leq M(1-z_j) \quad (j = 1, 2, \dots, p), \tag{5.1}$$

where M was set to 1000 in all instances of our experiments. Similarly, the implications (4.5) were represented with the big-M method,

$$-Mz_j \leq a_j \leq Mz_j \quad (j = 1, 2, \dots, p). \tag{5.2}$$

However, we had difficulty in finding a unified value of M for constraints (5.2) such that MISDO computation was not aborted due to numerical instability. Hence, we tuned the values of M through preliminary experiments as shown in Table 4.

These computations were performed on a Linux computer with an Intel Core2 Quad CPU (2.66 GHz) and 4 GB memory. MISDO problems were solved by using SCIP-SDP-2.0.0 [13] (http://www.opt.tu-darmstadt.de/scipsdp/) combined with SCIP 3.2.0 [1] (http://scip.zib.de/) and SDPA 7.3.8 [37] (http://sdpa.sourceforge.net/). The computation for solving the MISDO problem was terminated if it did not finish by itself within 10000 s. In this case, the best feasible solution obtained within 10000 s was taken as the result.

Table 4: Values of big-M for constraints (5.2)

				M	
Instance	n	p	Method	$(\kappa = 100)$	$(\kappa = 225)$
`Servo`	167	19	MISDO	2.25	2.25
			MISDO_{NE}	2.50	2.50
`AutoMPG`	392	25	MISDO	2.50	5.00
			MISDO_{NE}	1.05	1.50

Table 5: Results of solving MISDO problems ($\kappa = 100$)

Instance	n	p	Method	R^2	Cond	$\lvert S\rvert$	Time (s)
`Servo`	167	19	MISDO	**0.75877**	78.4	15	60.19
			MISDO_{NE}	**0.75877**	78.3	15	17.74
`AutoMPG`	392	25	MISDO	0.87429	76.9	19	>10000.00
			MISDO_{NE}	**0.87430**	92.8	21	5563.34

Table 6: Results of solving MISDO problems ($\kappa = 225$)

Instance	n	p	Method	R^2	Cond	$\lvert S\rvert$	Time (s)
`Servo`	167	19	MISDO	**0.75877**	97.1	15	4.99
			MISDO_{NE}	**0.75877**	104.1	15	5.49
`AutoMPG`	392	25	MISDO	0.87438	142.0	21	>10000.00
			MISDO_{NE}	**0.87438**	184.3	22	336.14

Tables 5 and 6 show the computational results of solving MISDO problems with $\kappa = 100$ and 225, respectively. The results for only the small-sized instances `Servo` and `AutoMPG` are shown in the tables because many of the large-scale MISDO problems could not be solved because of numerical instability* (i.e., violation of Slater's condition).

Tables 5 and 6 show that all the MISDO problems for `Servo` were solved within 61 s, and they were solved faster when $\kappa = 225$ than when $\kappa = 100$. Moreover, the normal-equation-based constraints worked effectively in speeding up the computations. For instance, in Table 5 the computation time of the MISDO formulation was reduced from 60.19 s to 17.74 s by incorporating the normal-equation-based constraints into the problem.

In the case of `AutoMPG`, only the computations of MISDO_{NE} finished within 10000 s for both $\kappa = 100$ and 225. Furthermore, MISDO_{NE} attained the largest R^2 value for every instance in Tables 5 and 6. These results demonstrate the effectiveness of the normal-equation-based constraints in the MISDO formulation for best subset selection to eliminate multicollinearity.

On the other hand, it is also the case that the computational performance of the MISDO approaches was much lower than that of the cutting plane algorithms. For instance in the case of `AutoMPG` with $\kappa = 225$, MISDO_{NE} took 336.14 s to solve the problem, but CPA_{BwE} required only 1.76 s to solve the same problem as shown in Table 3. These results

*Using SCIP-SDP 2.1.0 instead of 2.0.0 and softening feasibility tolerances of SCIP and SCIP-SDP solvers (`feastol` and `sdpsolverfeastol`, respectively) from 10^{-6} (default) to 10^{-4} could resolve numerical issues, but it missed true optimal solutions to some of the instances. Hence, we used SCIP-SDP 2.0.0 with its default parameters.

imply that our MISDO approach was not effective for the subset selection problem at the current moment; however, since the computational performance of MISDO algorithms is being improved, our MISDO formulation will work better in the future.

6. Conclusions

This paper addressed selection of the best subset of explanatory variables subject to an upper bound on the condition number for eliminating multicollinearity from linear regression models. To this end, we first developed a cutting plane algorithm in which valid inequalities for approximating the condition number constraint are iteratively added to the relaxed MIQO problem. We also devised an MISDO formulation for subset selection by transforming the condition number constraint into positive semidefinite constraints.

One contribution of this research is the establishment of a high-performance cutting plane algorithm. In particular, our algorithm strengthens valid inequalities by effectively using backward elimination and thus reduces the number of MIQO problems to be solved. A second contribution is a novel computational framework for eliminating multicollinearity based on MISDO formulation. This framework reformulates the subset selection problem as a single MISDO problem.

We found that our cutting plane algorithm frequently provided a better subset of variables than did the common local search algorithms. This finding demonstrates the effectiveness of the MIO approach in eliminating multicollinearity from a linear regression model. One limitation is that our MISDO formulation could be applied to only small-sized instances (e.g., $p \leq 26$); however, numerical techniques for solving MISDO problems are still in an early phase of development. This paper provides a new statistical application of MISDO formulation, and we hope that it will stimulate further research on numerical solutions for MISDO problems.

A future direction of study will be to use other indicators for detecting multicollinearity. For instance, the authors of this paper recently proposed MIQO formulations for best subset selection under the upper bound constraints on the variance inflation factor [31]. It is also possible in a cutting plane algorithm to check for the presence of multicollinearity by using the variance inflation factor.

Acknowledgments

The authors would like to thank Tristan Gally, TU Darmstadt for helpful advices on numerical issues of SCIP-SDP. This work was partially supported by JSPS KAKENHI Grant Number JP26560165.

References

[1] T. Achterberg: SCIP: Solving constraint integer programs. *Mathematical Programming Computation*, **1** (2009), 1–41.

[2] F.A. Al-Khayyal and J.E. Falk: Jointly constrained biconvex programming. *Mathematics of Operations Research*, **8** (1983), 273–286.

[3] E.M.L. Beale: Two transportation problems. In G. Kreweras and G. Morlat (eds.): *Proceedings of the Third International Conference on Operational Research* (Dunod, Paris and English Universities Press, London, 1963), 780–788.

[4] E.M.L. Beale and J.A. Tomlin: Special facilities in a general mathematical programming system for non-convex problems using ordered sets of variables. In J. Lawrence

(ed.): *Proceedings of the Fifth International Conference on Operational Research* (Tavistock Publications, London, 1970), 447–454.

[5] D. Bertsimas and A. King: OR forum–An algorithmic approach to linear regression. *Operations Research*, **64** (2016), 2–16.

[6] D. Bertsimas, A. King, and R. Mazumder: Best subset selection via a modern optimization lens. *The Annals of Statistics*, **44** (2016), 813–852.

[7] S. Chatterjee and A.S. Hadi: *Regression Analysis by Example, Fifth Edition* (Wiley, Hoboken, 2012).

[8] I.G. Chong and C.H. Jun: Performance of some variable selection methods when multicollinearity is present. *Chemometrics and Intelligent Laboratory Systems*, **78** (2005), 103–112.

[9] A.V. Dorugade and D.N. Kashid: Variable selection in linear regression based on ridge estimator. *Journal of Statistical Computation and Simulation*, **80** (2010), 1211–1224.

[10] M.A. Efroymson: Multiple regression analysis. In A. Ralston and H.S. Wilf (eds.): *Mathematical Methods for Digital Computers* (Wiley, New York, 1960), 191–203.

[11] D.E. Farrar and R.R. Glauber: Multicollinearity in regression analysis: The problem revisited. *The Review of Economic and Statistics*, **49** (1967), 92–107.

[12] L.E. Frank and J.H. Friedman: A statistical view of some chemometrics regression tools. *Technometrics*, **35** (1993), 109–135.

[13] T. Gally, M.E. Pfetsch, and S. Ulbrich: A framework for solving mixed-integer semidefinite programs. Optimization Online (2016). `http://www.optimization-online.org/DB_HTML/2016/04/5394.html`.

[14] R.F. Gunst and J.T. Webster: Regression analysis and problems of multicollinearity. *Communications in Statistics–Theory and Methods*, **4** (1975), 277–292.

[15] A.S. Hadi and R.F. Ling: Some cautionary notes on the use of principal components regression. *The American Statistician*, **52** (1998), 15–19.

[16] A.E. Hoerl and R.W. Kennard: Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, **12** (1970), 55–67.

[17] R.A. Horn and C.R. Johnson: *Matrix Analysis, Second Edition* (Cambridge University Press, New York, 2012).

[18] N.H. Jadhav, D.N. Kashid, and S.R. Kulkarni: Subset selection in multiple linear regression in the presence of outlier and multicollinearity. *Statistical Methodology*, **19** (2014), 44–59.

[19] I.T. Jolliffe: A note on the use of principal components in regression. *Applied Statistics*, **31** (1982), 300–303.

[20] K. Kimura and H. Waki: Minimization of Akaike's information criterion in linear regression analysis via mixed integer nonlinear program. arXiv preprint, arXiv:1606.05030 (2016).

[21] H. Konno and Y. Takaya: Multi-step methods for choosing the best set of variables in regression analysis. *Computational Optimization and Applications*, **46** (2010), 417–426.

[22] H. Konno and R. Yamamoto: Choosing the best set of variables in regression analysis using integer programming. *Journal of Global Optimization*, **44** (2009), 273–282.

[23] M. Lichman: UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science (2013). http://archive.ics.uci.edu/ml.

[24] E.R. Mansfield and B.P. Helms: Detecting multicollinearity. *The American Statistician*, **36** (1982), 158–160.

[25] W.F. Massy: Principal components regression in exploratory statistical research. *Journal of the American Statistical Association*, **60** (1965), 234–256.

[26] G.P. McCormick: Computability of global solutions to factorable nonconvex programs: Part I–Convex underestimating problems. *Mathematical Programming*, **10** (1976), 147–175.

[27] R. Miyashiro and Y. Takano: Subset selection by Mallows' C_p: A mixed integer programming approach. *Expert Systems with Applications*, **42** (2015), 325–331.

[28] R. Miyashiro and Y. Takano: Mixed integer second-order cone programming formulations for variable selection in linear regression. *European Journal of Operational Research*, **247** (2015), 721–731.

[29] T. Sato, Y. Takano, R. Miyashiro, and A. Yoshise: Feature subset selection for logistic regression via mixed integer optimization. *Computational Optimization and Applications*, **64** (2016), 865–880.

[30] T. Sato, Y. Takano, and R. Miyashiro: Piecewise-linear approximation for feature subset selection in a sequential logit model. *Journal of the Operations Research Society of Japan*, **60** (2017), 1–14.

[31] R. Tamura, K. Kobayashi, Y. Takano, R. Miyashiro, K. Nakata, and T. Matsui: Mixed integer quadratic optimization formulations for eliminating multicollinearity based on variance inflation factor. Optimization Online (2016). http://www.optimization-online.org/DB_HTML/2016/09/5655.html.

[32] R. Tibshirani: Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B (Methodological)*, **58** (1996), 267–288.

[33] M.J. Todd: Semidefinite optimization. *Acta Numerica*, **10** (2001), 515–560.

[34] L. Vandenberghe and S. Boyd: Semidefinite programming. *SIAM Review*, **38** (1996), 49–95.

[35] H. Wold: Estimation of principal components and related models by iterative least squares. In P.R. Krishnaiaah (ed.): *Multivariate Analysis* (Academic Press, New York, 1966), 391–420.

[36] S. Wold, A. Ruhe, H. Wold, and W.J. Dunn III: The collinearity problem in linear regression. The partial least squares (PLS) approach to generalized inverses. *SIAM Journal on Scientific and Statistical Computing*, **5** (1984), 735–743.

[37] M. Yamashita, K. Fujisawa, M. Fukuda, K. Kobayashi, K. Nakata, and M. Nakata: Latest developments in the SDPA family for solving large-scale SDPs. In M.F. Anjos and J.B. Lasserre (eds.): *Handbook on Semidefinite, Conic and Polynomial Optimization* (Springer, New York, 2012), 687–713.

[38] H. Zou and T. Hastie: Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, **67** (2005), 301–320.

Yuichi Takano
School of Network and Information
Senshu University
2-1-1 Higashimita, Tama-ku, Kawasaki-shi
Kanagawa 214-8580, Japan
E-mail: ytakano@isc.senshu-u.ac.jp