Subset Selection for Multiple Linear Regression via Optimization

Young Woong Park *[1] and Diego Klabjan †[2]

[1]Ivy College of Business, Iowa State University, Ames, IA, USA
[2]Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL, USA

January 14, 2020

Abstract

Subset selection in multiple linear regression aims to choose a subset of candidate explanatory variables that tradeoff fitting error (explanatory power) and model complexity (number of variables selected). We build mathematical programming models for regression subset selection based on mean square and absolute errors, and minimal-redundancy-maximal-relevance criteria. The proposed models are tested using a linear-program-based branch-and-bound algorithm with tailored valid inequalities and big M values and are compared against the algorithms in the literature. For high dimensional cases, an iterative heuristic algorithm is proposed based on the mathematical programming models and a core set concept, and a randomized version of the algorithm is derived to guarantee convergence to the global optimum. From the computational experiments, we find that our models quickly find a quality solution while the rest of the time is spent to prove optimality; the iterative algorithms find solutions in a relatively short time and are competitive compared to state-of-the-art algorithms; using ad-hoc big M values is not recommended.

Keywords. multiple linear regression, subset selection, high dimensional data, mathematical programming, linearization

1 Introduction

The multiple linear regression problem is a statistical methodology for predicting values of response (dependent) variables from a set of multiple explanatory (independent) variables by investigating the linear relationships among the variables. Given a fixed set of explanatory variables, the coefficients of the multiple linear regression model are estimated by minimizing the fitting error, where the standard setting uses the sum of squared errors (SSE) for measuring the fitting error. The subset selection problem, also referred to as variable selection or model selection, for multiple linear regression is to choose a subset of explanatory variables to build an efficient linear regression model. In detail, given a dataset with n observations and m explanatory variables, a subset of explanatory variables are used to build a regression model, where the goal is to decrease p, the number of explanatory variables in the model, as much as possible while maintaining error loss relatively small.

For selecting a subset of explanatory variables, an objective function is defined to measure the efficiency of the model [24]; the objective function is typically defined based on balancing the number of explanatory variables used and the fitting error. Criteria such as the mean square error (MSE), mean absolute error (MAE), adjusted r^2, Mallow's C_p, etc, are in this category for multiple linear regression and there are several works studying the L_0-norm-based feature selection in non-regression context [6, 30, 40]. There also exist objective functions balancing the magnitudes of the regression coefficients and the fitting errors; instead of the number of explanatory variables (non-zero coefficients), the regression coefficients are directly penalized. Among many variants in this category, ridge [19] and least absolute shrinkage and selection operator (LASSO) [38] regressions are the most popular models in multiple linear regression and there also exist recent papers studying the L_1-norm-based feature selection in a non-regression context [13, 20]. There also exist objective

*ywpark@iastate.edu
†d-klabjan@northwestern.edu

functions to select variables based on mutual information gain instead of minimizing fitting error. One of the popular criteria in this category is minimum-redundancy-maximum-relevance (mRMR) proposed by Ding and Peng [12] and Peng *et al.* [28]. Among various objective functions for selecting a subset, we focus on MAE, MSE, and mRMR in this paper.

Given a subset of explanatory variables, if SSE is minimized, an explicit formula is available for obtaining the optimal coefficients. On the other hand, when minimizing the sum of absolute errors (SAE), there is no explicit formula available. For this case, a linear program (LP) [9, 39] or iterative reweighted least squares algorithm [34] can be used to build the regression model.

When subset selection is required, algorithms for optimizing MSE have already been extensively studied. Among them, stepwise-type algorithms are frequently used in practice due to their computational simplicity and efficiency. An exact algorithm is to enumerate all possible regression models, but the computational cost is excessive. To overcome this computational difficulty, Furnival and Wilson [14] proposed a branch-and-bound algorithm, called *leaps-and-bound*, to find the best subset for MSE without enumerating all possible subsets. Miyashiro and Takano [26] proposed a mathematical programming model to maximize adjust r^2, which is equivalent to minimizing MSE. Given a fixed p, Bertsimas *et al.* [3] and Bertsimas and King [4] minimize SSE using mixed integer program (MIP)-based algorithms. For subset selection of least absolute deviation regression, Konno and Yamamoto [21] presented an MIP to optimize SAE given fixed p. Bertsimas *et al.* [3] proposed an MIP based algorithm for optimizing SSE and SAE given fixed p. A discrete first order method is proposed and used to warmstart the MIP formulation, which is formulated based on specially ordered sets [1], to avoid the use of big M. Bertsimas and King [4] proposed an MIP based algorithm for minimizing penalized SSE given fixed p. For a detailed review of algorithms for subset selection, the reader is referred to Miller [25].

Selecting a subset of explanatory variables with non-zero regression coefficients can be compared to general optimization problems with cardinality constraints. While several early works (e.g., Bienstock [5], de Farias and Nemhauser [10]) study general optimization problems with cardinality constraints from the optimization theoretical point of view, recent work directly focuses on mathematical programming models for regression subset selection. The MIP models in Bertsimas and Shioda [2], Bertsimas *et al.* [3], and Konno and Yamamoto [21] assume fixed p and the cardinality constraint is explicit in the models. Their models are distinguished by the objective functions and how they formulate subset selection; Konno and Yamamoto [21] optimized SAE by introducing binary variables, Bertsimas *et al.* [3] optimized SSE or SAE, Bertsimas and Shioda [2] optimized SSE without introducing binary variables. In contrast to the models in Bertsimas and Shioda [2], Bertsimas *et al.* [3], and Konno and Yamamoto [21], we optimize MSE and MAE without fixing p. Miyashiro and Takano [26] proposed a mathematical programming model to maximize adjust r^2, which is equivalent to minimize MSE. To the best of authors' knowledge, the model in Miyashiro and Takano [26] is the only mathematical programming model that directly maximizes adjust r^2 (equivalent to minimizing MSE), but there is no mathematical programming model directly optimizing mRMR or MAE.

Basic multiple linear regression analyses require a data matrix with $n > m + 1$; i.e., the number of observations must be greater than the number of explanatory variables plus one. Otherwise, the $n - 1$ linearly independent explanatory variables and one intercept variable yield a regression model with zero fitting error given a full rank data matrix. However, in practice, it is not that uncommon to have a data set with $m \geq n - 1$. For example, gene information has many attributes (explanatory variables) while only a few observations are usually available. In statistics, subset selection when $m \geq n$ is called *high dimensional variable selection*. Note that, if each row of the data matrix is an observation, the length of the data matrix is greater than the width when $m < n$, and the width of the data matrix is greater when $m \geq n$. Based on the shape of the data matrix, we hereafter refer to the cases $m < n$ and $m \geq n$ as *thin* and *fat cases*, respectively. For the fat case, Stodden [36] studied how model selection algorithms behave with a different but fixed ratio of $\frac{m}{n}$. Candes and Tao [7] proposed an l_1-regularized problem based approach, called *Dantzig selector*. However, their approach does not explicitly take into account the number of selected variables, which is different from our models.

Our contributions are as follows.

1. We present mathematical programs for the subset selection problem that directly minimize the popular criteria MAE and MSE. To the best of our knowledge, the proposed model for MAE is the first mathematical programming formulation that directly optimizes MAE. The proposed model for MSE is an equivalent model to the model of Miyashiro and Takano [26] which optimizes a different

objective function; our work has been conducted simultaneously with Miyashiro and Takano [26]. In the computational experiment, we observe that the proposed models quickly return a good candidate solution when solved by a commercial optimization solver.

2. We propose the first mathematical programming formulation that directly optimizes mRMR for the thin case, which also can be used for the fat case with trivial modifications. A modified version of the model is also proposed to balance mRMR and the fitting errors. The modified version integrates the mRMR-based feature selection and regression model building steps to obtain a model considering both mRMR and the error-based objective MAE or MSE. The computational experiment shows that the proposed models return different subsets from the MSE, MAE, and mRMR models in a relatively short computational time.

3. For the proposed mathematical programs for MSE and MAE, we propose exact and heuristic approaches to obtain big M values. The performances of the models with different big M values are discussed in the computational experiment. Further, the performance of the proposed big M-based formulations are compared with alternative mathematical programming formulations and implementations.

4. To overcome computational difficulties of the MIP models, we propose an iterative algorithm that gives a quality solution in a relatively short computational time for the fat case. We show that the algorithm yields a local optimal and we propose a randomized version of the algorithm to guarantee convergence to the global optimum. The computational experiment shows that the proposed algorithms are competitive compared to the state-of-the-art benchmarks.

The structure of the paper is as follows. In Section 2, the mathematical models for the thin case with MAE, MSE, and mRMR objectives are derived. In Section 3, for the fat case, we propose the iterative algorithm based on the mathematical models and derive the randomized version of the algorithm with the convergence result. Finally, we present computational experiments in Section 4.

2 Mathematical Models for Thin Case $(m < n)$

In this section, we derive mathematical programs to directly optimize MAE, MSE, and mRMR for the thin case. Throughout this paper, the following notation is used:

n : number of observations
m : number of explanatory variables
p : number of selected explanatory variables
$I = \{1, \cdots, n\}$: index set of observations
$J = \{1, \cdots, m\}$: index set of explanatory variables
$a = [a_{ij}] \in \mathbb{R}^{n\times m}$: data matrix corresponding to the independent variables
$a_j \in \mathbb{R}^n$: independent variable $j \in J$
$b = [b_i] \in \mathbb{R}^n$: data vector corresponding to the dependent variable.
ρ_{jk}: absolute sample correlation between explanatory variables $j, k \in J$
ρ_j: absolute sample correlation between explanatory variable $j \in J$ and the dependent variable

For all mathematical models derived, the following decision variables are used:

x_j: coefficient of the j^{th} explanatory variable, $j \in J$
y: intercept of the regression model
t_i: error term of the i^{th} observation, $i \in I$

$$z_j = \begin{cases} 1 & \text{if explanatory variable } x_j \text{ is included in the model} \\ 0 & \text{otherwise} \end{cases}, \quad j \in J\,.$$

Note that the multiple linear regression model takes the form $b_i = y + \sum_{j\in J} a_{ij}x_j + t_i$, for $i \in I$. Let us consider a regression model with fixed subset $\hat{S}$ of J. For the minimization of SAE given $\hat{S}$, the following

LP gives optimal regression coefficients:

$$\min \sum_{i \in I} \bar{t}_i \quad \text{s.t.} \quad t_i = \sum_{j \in \hat{S}} a_{ij} x_j + y - b_i, -\bar{t}_i \leq t_i \leq \bar{t}_i, \bar{t}_i \geq 0, i \in I. \tag{1}$$

We later use this LP as a subroutine when we need to construct a regression model that minimizes SAE given a fixed subset. Next we review the three subset selection criteria, which we use for the mathematical programming formulations. In the followings, SSE and SAE are taken with respect to a subset $\hat{S}$ of cardinality p.

1. MSE is one of the most popular criteria [37], defined as $\frac{SSE}{n-1-p}$. By minimizing MSE, we can balance SSE and p because SSE decreases in p. Another popular criteria is adjusted r^2, defined as $r_a^2 = 1 - \frac{MSE}{SST/(n-1)}$, where SST is the total sum of squares. Because $\frac{SST}{n-1}$ is a constant, maximizing r_a^2 is equivalent to minimizing MSE. This explains the equivalence of our model and Miyashiro and Takano [26].

2. MAE, defined as $\frac{SAE}{n-1-p}$, is an alternative to MSE for reducing the effect of outliers. Note that MAE is defined similarly to MSE, where SAE is used instead of SSE. MAE is a widely used criterion that is less sensitive to outliers and can also be used as an evaluation criterion when the model is fitted using squared errors [17]. For a detailed discussion of MAE compared with MSE, the reader is referred to Chai and Draxler [8] and Willmott and Matsuura [41]

3. mRMR, defined as $\frac{1}{p}\sum_{j \in \hat{S}} \rho_j - \frac{1}{p^2}\sum_{j,k \in \hat{S}} \rho_{jk}$, is frequently used to select features prior to running statistical models. By maximizing mRMR, the highly correlated explanatory variables to the dependent variable are selected (the first term in the expression) while maintaining the variables that are far away from each other (the second term in the expression).

We remark that the first objective is one of the most popular criteria practitioners use for selecting a subset, the second objective is a variant of the first, which is mainly concerned with reducing the effect of outliers, and the last objective is useful for screening the explanatory variables in an extreme fat case data.

2.1 Mean Square and Absolute Errors

In this section, we derive mathematical programs for MAE and MSE in Sections 2.1.1 and 2.1.2. For the proposed models, valid values for big M, which is an upper bound for the regression coefficients, and valid inequalities are derived in Sections 2.1.3 and 2.1.4.

2.1.1 Minimization of MAE

Observe that $MAE = \frac{SAE}{n-1-p}$ has two terms (SAE and p) that can be written as $SAE = \sum_{i \in I} |t_i|$ and $p = \sum_{j \in J} z_j$ in terms of the decision variables. Using these expressions, we can write a mathematical model

$$\begin{aligned}
\min \quad & \frac{\sum_{i \in I} |t_i|}{n-1-\sum_{j \in J} z_j} && && \text{(2a)}\\
s.t. \quad & t_i = \sum_{j \in J} a_{ij} x_j + y - b_i, && i \in I, && \text{(2b)}\\
& -Mz_j \leq x_j \leq Mz_j, && j \in J, && \text{(2c)}\\
& z_j \in \{0,1\}, t, x, y \text{ unconstrained.} && && \text{(2d)}
\end{aligned}$$

to minimize MAE. Observe that, if we add constraint $\sum_{j \in J} z_j = p$ to (2) given fixed p, we obtain an easier problem, which is equivalent to the model presented by Konno and Yamamoto [21] since the denominator of the objective becomes constant. By adding cardinality constraint with fixed p and by replacing (2c) with specially order sets based constraints, we obtain the model presented in Bertsimas *et al.* [3]. The remaining development is completely different from the work in Konno and Yamamoto [21] or Bertsimas *et al.* [3] and thus new. This is due to the fact that they assume fixed p which implies that model (2) is already linear. In our case we have to linearize this model which is not a trivial task. Note that (2) is a Mixed Integer Linear

Fraction Programming (MIFLP). There are numerous studies discussing solving MIFLP problems in the original form without linearizing the objective function, which is different from our approach linearizing the objective function to reformulate (2). The readers are referred to Schaible and Shi [32] and Stancu-Minasian [33] for detailed reviews of fractional programming literature.

Note that M in (2c) is a constant, which is an upper bound for x_j's, that we have not yet specified. Konno and Yamamoto [21] set an arbitrary large value for M in their study. For now, let us assume that a proper value of M is given (we derive a valid value for M in a later section). To linearize nonlinear objective (2a), we introduce

$$u = \frac{\sum_{i\in I}|t_i|}{n-1-\sum_{j\in J} z_j}. \tag{3}$$

Observe that u explicitly represents MAE. We linearize objective function (2a) by adding (3) as a constraint and setting u as the objective function. Then, (2) can be rewritten as

$$\begin{aligned}
\min \quad & u && && \text{(4a)}\\
s.t. \quad & \textstyle\sum_{i\in I}|t_i| = (n-1)u - u\sum_{j\in J} z_j, && && \text{(4b)}\\
& t_i = \textstyle\sum_{j\in J} a_{ij}x_j + y - b_i, && i\in I, && \text{(4c)}\\
& -Mz_j \le x_j \le Mz_j, && j\in J, && \text{(4d)}\\
& u\ge 0, z_j\in\{0,1\}, t,x,y \text{ unconstrained.} && && \text{(4e)}
\end{aligned}$$

In order to linearize nonlinear constraint (4b), we introduce $v_j = uz_j$, $j\in J$, which can be linearized using standard linearization techniques . Using a linearization technique [15] with proper settings, we obtain

$$\begin{aligned}
\min \quad & u && && \text{(5a)}\\
s.t. \quad & \textstyle\sum_{i\in I}|t_i| = (n-1)u - \sum_{j\in J} v_j && && \text{(5b)}\\
& t_i = \textstyle\sum_{j\in J} a_{ji}x_j + y - b_i, && i\in I, && \text{(5c)}\\
& -Mz_j \le x_j \le Mz_j, && j\in J, && \text{(5d)}\\
& v_j \le u, && j\in J, && \text{(5e)}\\
& u - M(1-z_j) \le v_j \le Mz_j, && j\in J, && \text{(5f)}\\
& v_j\ge 0, u\ge 0, z_j\in\{0,1\}, t,x,y \text{ unconstrained.} && && \text{(5g)}
\end{aligned}$$

Observe that we use M again in (5f) and a proper value for M is derived in a later section. We conclude that (5) is a valid formulation for (4) by the following proposition.

Proposition 1. An optimal solution to model (4) and an optimal solution to model (5) have the same objective function value.

The proof is given in Appendix A and is based on the fact that feasible solutions to (4) and (5) map to each other. Observe that the signs of $t, x,$ and y in (5) are not restricted. In order to make all variables non-negative, we introduce x_j^+, x_j^-, y^+ and y^-, in which $x_j = x_j^+ - x_j^-$ and $y = y^+ - y^-$. We also use t_i^+ and t_i^-, where $t_i = t_i^+ - t_i^-$, to replace the absolute value function in (5b). Finally, we obtain mixed integer program (6) for regression subset selection with the MAE objective.

$$\begin{aligned}
\min \quad & u && && \text{(6a)}\\
s.t. \quad & \textstyle\sum_{i=1}^{n}(t_i^+ + t_i^-) = (n-1)u - \sum_{j\in J} v_j, && && \text{(6b)}\\
& t_i^+ - t_i^- = \textstyle\sum_{j=1}^{m} a_{ij}(x_j^+ - x_j^-) + (y^+ - y^-) - b_i, && i\in I, && \text{(6c)}\\
& x_j^+ \le Mz_j, && j\in J, && \text{(6d)}\\
& x_j^- \le Mz_j, && j\in J, && \text{(6e)}\\
& v_j \le u, && j\in J, && \text{(6f)}\\
& u - M(1-z_j) \le v_j \le Mz_j, && j\in J, && \text{(6g)}
\end{aligned}$$

$$x_j^+ \geq 0, x_j^- \geq 0, y^+ \geq 0, y^- \geq 0, v_j \geq 0, u \geq 0, t_i^+ \geq 0, t_i^- \geq 0, z_j \in \{0,1\} \tag{6h}$$

It is known that either t_i^+ or t_i^- is equal to 0 if $\sum_{i\in I} |t_i|$ is minimized in the objective function. However, since $\sum_{i\in I} |t_i|$ is not directly minimized and binary variables are present in (5), we give the following proposition in order to make sure that (5) is equivalent to (6), where the proof is given in Appendix A.

Proposition 2. An optimal solution to (6) must have either $t_i^+ = 0$ or $t_i^- = 0$ for every $i \in I$.

By Proposition 2, it is easy to see that (6b) is equivalent to (5b). Therefore, (6) correctly solves (2). A final remark regarding the model is with regard to the dimension of the formulation. For a dataset with m candidate explanatory variables and n observations, formulation (6) has $2n + 4m + 3$ variables (including m binary variables) and $n + 5m + 1$ constraints (excluding non-negativity constraints).

2.1.2 Minimization of MSE

In this section, we derive a quadratically constrained mixed integer programming model based on the results in Section 2.1.1, which gives an equivalent formulation to Miyashiro and Takano [26] as maximizing adjusted r^2 is equivalent to minimizing MSE. Our work has been conducted simultaneously with Miyashiro and Takano [26].

Observe that the only difference between MSE and MAE is that MSE has $\sum_{i=1}^n t_i^2$, whereas MAE has $\sum_{i=1}^n |t_i|$. Hence, the left hand side of (6b) is replaced by $\sum_{i=1}^n (t_i^+ - t_i^-)^2$. Also, in order to make the constraint convex, we use inequality instead of equality. Hence, we use

$$\sum_{i\in I} (t_i^+ - t_i^-)^2 \leq (n-1)u - \sum_{j\in J} v_j \tag{7}$$

instead of (6b). Finally, the mixed integer quadratically constrained program with the convex relaxation reads

$$\min\{u | (7), (6c) - (6h)\}. \tag{8}$$

Note that we use inequality in (7) to have the convex constraint, but u is correctly defined only when (7) is at equality. Hence, we need the following proposition.

Proposition 3. An optimal solution to (8) must satisfy (7) at equality.

The proof is given in Appendix A. By Proposition 3, we know that (7) is satisfied at equality at an optimal solution, hence (8) correctly solves the problem.

2.1.3 Big M for x_j's and v_j's

Deriving a tight and valid value of M in (6) and (8) is crucial for two reasons. For optimality, too small values cannot guarantee optimality even when the optimization model is solved optimally. For computation, a large value of M causes numerical instability and slows down the branch-and-bound algorithm. Recall that we assume that a valid value of M is given for the formulations (6) and (8) and that the same notation M is used for both x_j's and v_j's. However, x_j's and v_j's are often in different magnitudes. Hence, it is necessary to derive distinct and valid values of M for x_j's and v_j's.

In this section, we derive valid values of M for x_j's and v_j's in (6). The result also holds for (8) with trivial modifications. Among the two exact approaches proposed in this section, the first approach is based on the logic similar to Bertsimas *et al.* [3], where a similar approach is provided without a validity check for a different problem minimizing SSE given a fixed p. We also provide a computationally faster procedure for M for x_j's in (8) as an alternative. Both of the M values do not cause any numerical problems in our experiments.

First, let us consider M for v_j's. Observe that a valid M for v_j must be greater than all possible values for u. However, it is generally better to have tight upper bounds. Hence, we use mae_m, the mean absolute error of an optimal regression model with all m explanatory variables, as upper bounds. We set

$$M := mae_m \tag{9}$$

for every v_j in (6). Note that (9) can be calculated by LP formulation (1) in polynomial time. By using the M value in (9), we treat regression models that have worse objective function values than mae_m as infeasible.

Next, let us consider M for x_j's in (6) for MAE. We start with the following assumption.

Assumption 1. Dataset $\{b, a_1, a_2, \cdots, a_m\}$ is linearly independent.

This assumption implies that there is no regression model with total error equal to 0 among all possible subsets of the m explanatory variables. This is a mild assumption because, in practice, we typically have a dataset with structural and random noises and it is unlikely to have zero error.

In order to find a valid value of M for x_j's in (6), we formulate an LP. Let μ be the decision variable having the role of M. Let $\bar{b} = \frac{\sum_{i \in I} b_i}{n}$ and $T_{max} = \sum_{i \in I} |b_i - \bar{b}|$ be the average of b_i's and the maximum total error bound allowed, respectively. Any attractive regression model should have the total error less than T_{max} in order to justify the effort of building a regression model, because $SAE > T_{max}$ with $p > 0$ gives an automatically worse objective function value than the model with no explanatory variable. This requirement is written as $\sum_{i \in I}(t_i^+ + t_i^-) \leq T_{max}$. Because for now we are only concerned with feasibility, we can ignore u and all related constraints and variables (6b), (6f), z_j's, and v_j's. Then, we have the following feasibility set:

$$\sum_{i \in I}(t_i^+ + t_i^-) \leq T_{max}, \tag{10a}$$

$$t_i^+ - t_i^- = \sum_{j \in J} a_{ij}(x_j^+ - x_j^-) + (y^+ - y^-) - b_i, \qquad i \in I, \tag{10b}$$

$$x_j^+ \leq \mu, \qquad j \in J, \tag{10c}$$

$$x_j^- \leq \mu, \qquad j \in J, \tag{10d}$$

$$\mu \geq 0, x_j^+ \geq 0, x_j^- \geq 0, y^+ \geq 0, y^- \geq 0, t_i^+ \geq 0, t_i^- \geq 0. \tag{10e}$$

For notational convenience, let $Y = (x^+, x^-, y^+, y^-, t^+, t^-, \mu)$ be a vector in (10).

Next, let us try to increase x_k^+ to its maximum value. For a fixed $0 < \varepsilon < 1$, we define the objective as

$$\max \quad x_k^+ - \varepsilon\mu.$$

With the second term, we force μ to be the maximum value we need, yet not preventing a further increment of x_k^+. From the linear program

$$\max\{x_k^+ - \varepsilon\mu | (10a)\text{-}(10e), x_k^- = 0\}, \tag{11}$$

we obtain $\hat{M}_k^+$, a candidate for M, from the value of μ of an optimal solution solution to (11). Similarly, $\hat{M}_k^-$ is obtained from $\max\{x_k^- - \varepsilon\mu | (10a)\text{-}(10e), x_k^+ = 0\}$. Then the maximum value for explanatory variable x_k can be obtained by setting $\hat{M}_k = \max\{\hat{M}_k^+, \hat{M}_k^-\}$. Finally, considering all explanatory variables, we define $\hat{M}$ as

$$\hat{M} = \max_{j \in J} \hat{M}_j. \tag{12}$$

Before we proceed, we first need to make sure that (11) is not unbounded so that the values are well defined.

Proposition 4. Linear program (11) is bounded.

Lemma 1. Let $\hat{M}$ be the value obtained from (12) and $\bar{Y} = (\bar{x}^+, \bar{x}^-, \bar{y}^+, \bar{y}^-, \bar{v}_j, \bar{u}, \bar{t}^+, \bar{t}^-, \bar{z})$ be a feasible solution of (6) with $\hat{M}$ and SAE less than or equal to T_{max}. Then, $\hat{Y} = (\bar{x}^+, \bar{x}^-, \bar{y}^+, \bar{y}^-, \bar{t}^+, \bar{t}^-, \hat{M})$ is a feasible solution for (10).

The proofs are given in Appendix A. Note that Lemma 1 implies that (10) covers all possible values of x_j^+ and x_j^- of (6) with the maximum total error bound T_{max}. Note also that $\hat{M}$ in (12) is the maximum value out of all possible values of x_j^+ and x_j^- that (10) covers.

Proposition 5. For all regression models with SAE less than or equal to $T_{\max}$, $\hat{M}$ in (12) is a valid upper bound for x_j^+'s and x_j^-'s in (6).

Proof. For a contradiction, suppose that $\hat{M}$ is not a valid upper bound for x_j's in (6). That is, there exists a regression model $(\bar{x}^+, \bar{x}^-, \bar{y}^+, \bar{y}^-)$ with total error less than T_{max} but $\bar{x}_q^+ > \hat{M}$, in which $\bar{x}_q^+$ is the coefficient for explanatory variable q. However, by Lemma 1, we must have a corresponding feasible solution $\bar{Y} = (\bar{x}^+, \bar{x}^-, \bar{y}^+, \bar{y}^-, \bar{t}^+, \bar{t}^-, \hat{M})$ for (10) with $\bar{x}_q^+ > \hat{M}$. Note that $\bar{Y}$ must satisfy $\bar{x}_q^+ \leq M_q^+$ from (10d). Then, $M_q^+ \geq x_q^+ > \hat{M}$ implies $M_q^+ > \hat{M}$. This contradicts definition (12). A similar argument holds if $\bar{x}_q^- > \hat{M}$. Hence, $\hat{M}$ is a valid upper bound. □

Observe that a similar approach can be used to derive a valid value of M for x_j's in (8) for MSE. Calculating a valid value M for x_j's in (6) and (8) consists of solving $2m$ LPs and $2m$ quadratically constrained convex quadratic programs (QCP). Hence, we conclude that it can be obtained in polynomial time.

To reduce the computational time for the big M calculation, we present an alternative approach that works for MSE models from a different perspective.

Note that we can obtain coefficients of an optimal regression model that minimizes SSE over all explanatory variables as $\hat{x} = (a^\top a)^{-1} a^\top b$, where $a \in \mathbb{R}^{n \times m}$ and $b \in \mathbb{R}^{n \times 1}$. This is equivalent to solving $Ax = B$, with $A = a^\top a \in \mathbb{R}^{m \times m}$ and $B = a^\top b \in \mathbb{R}^{m \times 1}$. For a rational number $r = \frac{r_{num}}{r_{den}}$ ($r_{num} \in \mathbb{Z}$, $r_{den} \in \mathbb{N}$, r_{num} and r_{den} relative prime numbers), a rational vector $B = [\beta_1, \cdots, \beta_m]$, and a rational matrix $A = [\alpha_{ij}]_{i=1,\cdots,m, j=1,\cdots,m}$, let us define

$$\begin{aligned}
&size(r) := 1 + \lceil \log_2(|r_{num}| + 1) \rceil + \lceil \log_2(r_{den} + 1) \rceil \\
&size(B) := \textstyle\sum_{i \in I} size(\beta_i) \\
&size(A) := m^2 + \textstyle\sum_{i \in I} \sum_{j \in J} size(\alpha_{ij}).
\end{aligned}$$

Note that it is known that the size of solutions to $Ax = B$ are bounded. Here, we extend this over the various submatrices of A and subvectors of B encountered in our subset selection procedure. The following proposition provides a valid value of M.

Proposition 6. Value $M := 2^{size(A)size(B)-1}$ is a valid upper bound for x_j^+'s and x_j^-'s in (8).

The proof of Proposition 6 and the omitted detailed derivations are available in Section 1 of the online supplement. Observe that $size(A)$ and $size(B)$ can be calculated in polynomial time. In detail, it takes $O(mnh)$ in which h is the number of digits of the largest absolute number among all elements of A and B to compute M. Recall that the previous approach requires to solve $2m$ QCPs. Hence, we have an alternative polynomial time big M calculation procedure which is computationally more efficient than the one provided by Proposition 5. However, this procedure yields a larger value of M.

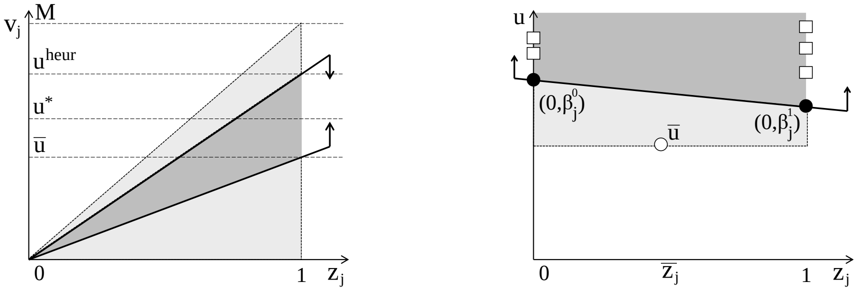
2.1.4 Valid Inequalities

To accelerate the computation, we apply several valid inequalities at the root node of the branch and bound algorithm. Let u^{heur} and $\bar{u}$ be the objective function values of a heuristic and the LP relaxation, respectively. Let β_j^0 (β_j^1) be the objective function value of the LP relaxation of (6) after fixing $z_j = 0$ ($z_j = 1$). Then, the following inequalities are valid for (6):

$$\begin{aligned}
&v_j \leq u^{heur} z_j, && j \in J && (13)\\
&v_j \geq \bar{u} z_j, && j \in J && (14)\\
&u \geq (\beta_j^1 - \beta_j^0) z_j + \beta_j^0 && j \in J && (15)
\end{aligned}$$

We do not provide proofs as it is trivial to establish their validity. In Figure 1, we illustrate the valid inequalities. In both figures, the dark and light-shaded areas represent the feasible and infeasible region, respectively, after applying the valid inequalities, whereas the combined area represents the original feasible region of the formulation. In Figure 1(a), valid inequalities (13) and (14) are presented. Value u^* is the optimal objective function value. In Figure 1(b), $\bar{u}$ is the objective function value of the LP relaxation with non-integer z_j before applying (15). The black circles represent $(0, \beta_j^0)$ and $(1, \beta_j^1)$ that give valid lower bounds for any integer solution. Observe that integer feasible solutions (empty rectangles in the figure) are in the feasible region after applying the valid inequality.

Note that (13) can be generated given an objective value of any feasible solution. For MAE, generating (14) and (15) requires solving one LP and two LPs, respectively, for each $j \in J$. For MSE, generating (14) and (15) requires solving one QCP and two QCPs, respectively.



(a) Valid inequalities (13) and (14)

(b) Valid inequality (15)

Figure 1: Illustration of the valid inequalities

2.2 Minimal-Redundancy-Maximal-Relevance

Given a and b, the mRMR criterion can be modeled as the following optimization problem:

$$\max_{S} \frac{1}{|S|}\sum_{j\in S}\rho_j - \frac{1}{|S|^2}\sum_{j,k\in S}\rho_{jk}. \tag{16}$$

In this section, we assume that we want to find set S with $|S| = p$, which is different from the previous treatment, i.e., here p is fixed. Using the binary variables z_j previously defined, (16) can be written as

$$\max\left\{\frac{p\sum_{j\in J}\rho_j z_j - \sum_{j,k\in J}\rho_{jk}z_j z_k}{p^2}\middle| z\in\{0,1\}^m, \sum_{j\in J} z_j = p\right\}. \tag{17}$$

By introducing new variable $z_{jk} \equiv z_j z_k$, (17) can be converted into an MIP as follows.

$$\begin{aligned}
\max \quad & \sum_{j\in J}\frac{\rho_j}{p}z_j - \sum_{j,k\in J}\frac{\rho_{jk}}{p^2}z_{jk} && && \text{(18a)}\\
s.t. \quad & z_{jk} \geq z_j + z_k - 1 && j,k\in J && \text{(18b)}\\
& \sum_{j\in J} z_j = p && && \text{(18c)}\\
& z_j\in\{0,1\}, z_{jk}\in\{0,1\} && && \text{(18d)}
\end{aligned}$$

This model is the first approach in the literature that guarantees global optimality for the mRMR criterion. However, if (18) is solved approximately, it may not improve the solution of the greedy algorithm, which is often used in practice. Further, if the selected features by (18) will be used for building a regression model, it is beneficial to consider the regression fit simultaneously by integrating the mRMR and the traditional error-based objectives. Hence, we propose to combine (6) and (18) to optimizing SAE while guaranteeing good objective function values for (18).

Let $\bar{\Omega}$ be the optimal objective function value of (16). With a fractional parameter $\lambda\in[0,1]$, the following constraint guarantees at most $\frac{\lambda}{100}\%$ away from $\bar{\Omega}$:

$$\tfrac{1}{|S|}\textstyle\sum_{j\in S}\rho_j - \tfrac{1}{|S|^2}\sum_{j,k\in S}\rho_{jk} \geq \bar{\Omega} - \text{sign}(\bar{\Omega})\cdot\lambda\cdot|\bar{\Omega}|.$$

Regardless of the sign of $\bar{\Omega}$, the lower bound is smaller than $\bar{\Omega}$ by $\lambda\cdot\bar{\Omega}$. Combining the constraint with the MIP model optimizing SAE, the following MIP is obtained.

$$\min \quad \textstyle\sum_{i\in I} t_i^+ + t_i^- \tag{19a}$$

$$s.t. \quad t_i^+ - t_i^- = \textstyle\sum_{j\in J} a_{ij}(x_j^+ - x_j^-) + y^+ - y^- - b_i, \qquad i \in I, \tag{19b}$$

$$-Mz_j \le x_j \le Mz_j, \qquad j \in J, \tag{19c}$$

$$\sum_{j\in J} \frac{\rho_j}{p} z_j - \sum_{j,k\in J} \frac{\rho_{jk}}{p^2} z_{jk} \ge \bar{\Omega} - \mathrm{sign}(\bar{\Omega}) \cdot \lambda \cdot |\bar{\Omega}|, \tag{19d}$$

$$z_{jk} \ge z_j + z_k - 1, \qquad j,k \in J, \tag{19e}$$

$$\sum_{j\in J} z_j = p, \tag{19f}$$

$$z_j \in \{0,1\}, z_{jk} \in \{0,1\}, t_i^+, t_i^-, x_j^+, x_j^-, y^+, y^- \ge 0 \tag{19g}$$

Note that (19) combines the mRMR feature selection and regression model building procedures, whereas (18) provides pre-screening of explanatory variables for the regression model building step in the next stage. A similar model for SSE can be obtained by replacing (19a) by $\sum_{i\in I}(t_i^+ + t_i^-)^2$, and (19) can be used for the fat case with trivial modifications.

3 Mathematical Models and Algorithms for Fat Case $(m \ge n)$

Let us consider the fat case, in which there are more explanatory variables than observations. A natural extension of (6) or (8) for the fat case is to add cardinality constraint $\sum_{j\in J} z_j \le n-2$. This constraint successfully selects a proper number of explanatory variables in many cases, however, we found that the objective MAE and MSE for the fat case could be problematic in some cases; the penalty on the number of explanatory variables by MAE or MSE is too weak (or strong) and the optimal solution selects $n-2$ (or 0) explanatory variables.

Minimizing SAE can be thought as approximating the right-hand side (dependent values b) using a combination of columns (explanatory variables). If we have more linearly independent explanatory variables than observations, we can always build a regression model with $SAE = 0$. Hence, if we allow $p \ge n-1$, then the MAE objective is not useful. Further, due to the definition of $MAE = \frac{SAE}{n-1-p}$, we must have $p \le n-2$ in order to make the numerator positive.

Suppose we can select $n-2$ explanatory variables out of m $(m > n-2)$ candidate explanatory variables. Because SAE converges to zero as we add more linearly independent explanatory variables and because $p = n-2$ and n are close to each other, SAE can be near zero. In this case, having $n-2$ explanatory variables might not be penalized enough by the definition of MAE. This could make $p = n-2$ optimal and it actually happens in many instances studied in Section 4, which is not a desired solution in most cases. Hence, even with the restriction $p \le n-2$, MAE may not be a useful criteria. In order to fix this issue, we use a slightly modified objective function by additionally penalizing having too many explanatory variables in the regression model:

$$MAE_a = \frac{SAE + \frac{p}{n-2} mae_0}{n-1-p}, \tag{20}$$

where $mae_0 = \frac{\sum_{i\in I}|b_i - \bar{b}|}{n-1}$ is the mean absolute error of the optimal regression model with $p = 0$. Observe that (20) is equivalent to MAE when $p = 0$. The penalty term increases as p increases. To optimize (20), (6) can be modified as

$$\min\{u | (6b) - (6e), (6h), \sum_{j\in J} z_j \le n-2, v_j \le u + \frac{mae_0}{n-2}, u + \frac{mae_0}{n-2} - M(1-z_j) \le v_j \le Mz_j\}. \tag{21}$$

For the detailed derivations and modifications, please consider Appendix C. Finally, we remark that all algorithms proposed in this section can also optimize MSE and MAE.

3.1 Core Set Algorithm

Observe that (21) might be difficult to solve optimally if the data is large because the number of binary variables increases as m increases. To overcome this computational difficulty and get a quality solution quickly, we develop an iterative algorithm based on (21) and the popular core set concept in computer science and operations research [16].

Let C be a subset of J such that $|C| \leq n-2$, with the cardinality of C defined by

$$\Theta = |C| = \min\{n\theta, n-2\}, \tag{22}$$

where $0 < \theta < 1$ is a fraction that defines the target cardinality of C. We refer to C as the *core set* and iteratively solve

$$\min\{u|(6b),(6c)-(6e),(6h),(28),(29)\} \tag{23}$$

that is obtained by dropping the cardinality constraint (30) from (21). Hereafter, we assume that (23) is always solved with C instead of J, with $|C| \leq n-2$ being ensured by (22).

We present the algorithmic framework in Algorithm 1 based on the core set concept. Let S^* be the current best subset in Algorithm 1 with corresponding objective function value mae^*_a. In Steps 1 - 3, we initialize core set C with cardinality not exceeding Θ. We solve (23) with C in Step 5 and then update C in Step 6. We iterate these steps until there is no improvement of the objective function value from a previous iteration. We remark that the worst case run time of Algorithm 1 is exponential because (23) is solved by the branch-and-bound algorithm, which has exponential worse case run time, in each iteration. However, in practice, Algorithm 1 terminates quickly as shown in the experimental results in Section 4.

Algorithm 1 Core-Heuristic

```
Input: $\theta$ (core set factor)
1: $\Theta \leftarrow \min\{n\theta, n-2\}$
2: $(S^*, mae^*_a) \leftarrow$ stepwise heuristic with $J$ and constraint $p \leq \Theta$
3: $(S^*, mae^*_a, C, \Theta) \leftarrow$ Update-Core-Set$(S^*, mae^*_a, \Theta)$
4: while objective function value is improving do
5:     $(S^*, mae^*_a) \leftarrow$ solve (23) with $C$
6:     $(S^*, mae^*_a, C, \Theta) \leftarrow$ Update-Core-Set$(S^*, mae^*_a, \Theta)$
7: end while
```

We next explain how the core set is updated. The updating algorithm is presented in Algorithm 2. In Steps 13 and 14, the idea is to keep the explanatory variables of the current best subset S^* in the core set and additionally selecting explanatory variables not in S^* based on scores T_j. The score is defined based on how much of the error could be reduced if we add explanatory variable j to the current best subset S^*. In Steps 1 - 6, we calculate T_j's and E^j_a's by checking neighboring subsets. Note that T_j's can be calculated by LP formulation (1). In Steps 7 - 12, we update the current best subset S^* if we found a better solution in Steps 1 - 6. If S^* is updated, we go to Step 1 and restart the algorithm with new S^* and Θ. Observe that E^j_a's in Steps 1 - 3 are only for updating S^* in Step 8, whereas T_j's and E^j_a's in Steps 4 - 6 are also used to define B in Step 13.

Let us define the neighborhood of set $\bar{S}$ as

$$\mathcal{N}(\bar{S}) = \{S \subset J || S \triangle \bar{S}| \leq 1\}, \tag{24}$$

where $S \triangle \bar{S}$ defines the symmetric difference of S and $\bar{S}$. Through the following propositions, we show that Algorithm 1 does not cycle and terminates with a local optimal solution based on the neighborhood definition given in (24).

Proposition 7. Algorithm 1 does not cycle.

For the proof, see Lemmas OS 6 and 7 (OS stands for online supplement), which guarantee that there is no cycle in the loop of Algorithm 1.

Proposition 8. Algorithm 1 gives a local optimum.

Proof. When Algorithm 1 terminates, all subsets that are neighbors to S^*, defined by (24), are evaluated in Steps 1 - 6 of Algorithm 2, but there is no better solution than S^*. Hence, Algorithm 1 gives a local optimum. □

```
Algorithm 2 Update-Core-Set
Input: $S^*$ (current best subset), $mae_a^*$ (current best obj value), $\Theta$ (core set cardinality)
Output: $S^*$ (new current best subset), $mae_a^*$ (new current best obj value), $C$ (new core set), $\Theta$ (new core set cardinality)
 1: for $j \in S^*$ do
 2:     $T_j \leftarrow$ SAE of subset $S^* \setminus \{j\}$, $E_a^j \leftarrow \frac{T_j + \frac{|S^*|-1}{n-2} mae_0}{n-1-|S^*|-1}$
 3: end for
 4: for $j \in J \setminus S^*$ do
 5:     $T_j \leftarrow$ SAE of subset $S^* \cup \{j\}$, $E_a^j \leftarrow \frac{T_j + \frac{|S^*|+1}{n-2} mae_0}{n-1-|S^*|+1}$
 6: end for
 7: if $\min_{j \in J} E_a^j < mae_a^*$
 8:     update $S^*$ to $T_j$ that gives minimum $E_a^j$ value
 9:     if $|S^*| = \Theta$ then $\Theta \leftarrow \min\{\Theta + 1, n-2\}$
10:     $mae_a^* \leftarrow \min_{j \in J} E_a^j$
11:     go to Step 1
12: end if
13: $B \leftarrow \{\Theta - |S^*|$ explanatory variables in $J \setminus S^*$ with smallest $T_j$'s $\}$
14: $C \leftarrow S^* \cup B$
```

3.2 Randomized Core Set Algorithm

We also present a randomized version of Algorithm 1, which we call *Core-Random*. By constructing a core set randomly and by executing the while loop of Algorithm 1 infinitely many times, we show that we can find a global optimal solution with probability 1 when $\theta = 1$. The randomized version of *Update-Core-Set* is presented in Algorithm 3. *Update-Core-Set-Random* is similar to *Update-Core-Set*, with one difference. Instead of the greedy approach in Steps 13-14 of Algorithm 2, we randomly choose $n-2$ explanatory variables one-by-one without replacement based on a probability distribution.

```
Algorithm 3 Update-Core-Set-Random
Input: $S^*$ (current best subset), $mae_a^*$ (current best obj value), $\Theta$ (core set cardinality)
Output: $S^*$ (new current best subset), $mae_a^*$ (new current best obj value), $C$ (new core set), $\Theta$ (new core set cardinality)
1: Steps 1 - 12 of Algorithm 2
2: Define initial probabilities based on (26)
3: $C \leftarrow \emptyset$, $\bar{J} \leftarrow J$
4: while $|C| < \Theta$
5:     Select explanatory variable $k$ in $\bar{J}$ based on generalized Bernoulli with probabilities $p_j$
6:     $C \leftarrow C \cup \{k\}$, $\bar{J} \leftarrow \bar{J} \setminus \{k\}$, renormalize $p_j$'s based on (27)
7: end-while
```

Let us next describe the initial probability distribution used in Step 2 of Algorithm 3. Let U_j be the current best objective function value whenever explanatory variable j is included in the regression model. We update U_j's at each iteration throughout the entire algorithm. In detail, we set $U_j := mae_a^*$ for $j \in S^*$ whenever current best objective function value mae_a^* and subset S^* are updated. In order to enhance the local optimal search, we give a bonus to the columns currently in S^* by setting weight $w_j = 0.5$ if $j \in S^*$ and $w_j = 1$ if $j \in J \setminus S^*$. Observe that giving the same weight for all $j \in J$ is equivalent to a random search. On the other hand, if the weight for S^* is much smaller (hence much greater selection probability) than the weight for $j \in J \setminus S^*$, then we are likely to choose all variables in S^*, which is similar to Algorithm 2. By means of a computational experiment, we found out that giving twice more weights for $j \in J \setminus S^*$ compared to $j \in S^*$ balances exploration and exploitation.

We normalize U_j's and generate $\bar{U}_j$'s so that $\min_{j\in J}\bar{U}_j = -0.5$ and $\max_{j\in J}\bar{U}_j = 0.5$. In detail,

$$\bar{U}_j = \frac{w_j U_j - \bar{U}_{mid}}{\bar{U}_{max} - \bar{U}_{min}} \quad \text{for } j \in J, \tag{25}$$

where $\bar{U}_{min} = \min_{j\in J} w_j U_j$, $\bar{U}_{max} = \max_{j\in J} w_j U_j$, and $\bar{U}_{mid} = (\bar{U}_{max} - \bar{U}_{min})/2$. Finally, we define probabilities using the exponential function

$$q_j = \frac{e^{-\bar{U}_j}}{\sum_{j\in J} e^{-\bar{U}_j}} \quad \text{for } j \in J. \tag{26}$$

From definitions (25) and (26), we have the following characteristic of q_j's.

Lemma 2. We have $\dfrac{\max_{j\in J} q_j}{\min_{j\in J} q_j} \leq 2.72$ for any values of q_j's.

The proof is available in Section 2 of the online supplement. By the lemma, we know that the best explanatory variables in S^* has at most 2.72 times higher chance than the worst explanatory variable to be picked. Observe that, once we select an explanatory variable in Step 5, we need to exclude the selected explanatory variable in the next selection iteration. This can be thought as sampling without replacement. Let $\bar{J}$ be the set of explanatory variables that have not been selected in the previous selection iterations. In Step 6, we add explanatory variable k to the core set and exclude it from $\bar{J}$. Then, we normalize the probability distribution based on

$$q_j = \frac{q_j}{\sum_{j\in \bar{J}} q_j} \quad \text{for } j \in \bar{J} \tag{27}$$

so that we only consider variables that have not been picked and the corresponding probabilities sum to 1. It is easy to see that q_j's after normalization by (27) are strictly greater than q_j's before normalization. Note also that q_j's in (27) also satisfy Lemma 2, since in (27) we are multiplying them by a constant.

Now we are ready to show that *Core-Random* with $\theta = 1$ finds a global optimal solution with probability 1. We first precisely review how *Core-Random* proceeds and define a detailed notation for the analysis. In iteration t, the following steps are performed.

1. We solve (23) with C in *Core-Random* and obtain S^*. Note that the core set is from the previous iteration. Hence, we denote the core set as C_{t-1}.
2. In Step 1 of *Update-Core-Set-Random*, we check the neighborhood of S^* obtained from (23) and update S^* if applicable.
3. After Step 1 of *Update-Core-Set-Random*, we obtain q_j's from (26). Let $q_j^{(t)}$ be the initial probability, defined in (26), used to construct the core set in iteration t.
4. In Step 2 of *Update-Core-Set-Random*, we construct core set C_t based on $q_j^{(t)}$'s. Note that C_t is used in iteration $t+1$ to solve (23).

Let S^{opt} be an optimal subset. If $S^{opt} \subset C_t$ for a core set C_t, then we can find a global optimal solution by solving (23) in iteration $t+1$. We first derive a lower bound of the probability for the event $S_{opt} \subset C_t$ given any previous iterations.

Lemma 3. Let $\mathcal{H}_{t-1}$ be the set that includes any collection of the events that have happened prior to iteration t. Then, we have

$$P[S^{opt} \subset C_t | \mathcal{H}_{t-1}] \geq \left(\frac{1}{1 + 2.72(m-1)}\right)^{\Theta}.$$

The proof is available in Section 2 of the online supplement. Let mae_a^{opt} be the optimal objective function value of (21) over the entire J and $mae_a(t)$ be the objective function value of the current best solution in iteration t of *Core-Random*, i.e., the objective value with respect to S^*. Let A_t be the event $\{S^{opt} \not\subset C_t\}$ in iteration t. For notational convenience, let

$$\varphi = \left(\tfrac{1}{1+2.72(m-1)}\right)^{\Theta}$$

be the lower bound for $P[S^{opt} \subset C_t | \mathcal{H}_{t-1}]$ from Lemma 3. Based on Lemma 3, we present the following lemmas with the proofs given in Section 2 of the online supplement.

Lemma 4. We have $P\Big[\bigcap_{k=1}^{t} A_k\Big] \leq (1-\varphi)^t$ for any iteration t.

Lemma 5. We have $P\Big[mae_a(t) = mae_a^{opt}\Big] \geq 1-(1-\varphi)^t$ for any iteration t.

Finally, we show that *Core-Random* finds a global optimal solution with probability 1 as iterations continue infinitely.

Proposition 9. We have $\lim_{t\to\infty} P\Big[mae_a(t) = mae_a^{opt}\Big] = 1$.

Proof. Since $0 < \varphi < 1$ by the definition of φ, we have $\lim_{t\to\infty}(1-\varphi)^t = 0$. Using this result, we derive

$$\lim_{t\to\infty} P\Big[mae_a(t) = mae_a^{opt}\Big] \geq \lim_{t\to\infty} 1-(1-\varphi)^t = 1.$$

Hence, we obtain $\lim_{t\to\infty} P\Big[mae_a(t) = mae_a^{opt}\Big] = 1$. □

4 Computational Experiment

In this section, we present computational experiments for all proposed models and algorithms in Section 2 and Section 3. They are compared to benchmark algorithms and to each other. To test the performance, we use randomly generated instances and a personal computer with 8 GB RAM and Intel Core i7 (2.40 GHz dual core) was used for the experiments in Section 4.3 and a server with Xeon 2.8 GHz CPU and 15GB RAM is used for all other experiments. All models and algorithms are implemented in C# and CPLEX.

4.1 Experimental Design

We obtained many publicly available instances for the subset selection problem. The majority of them were very easy to solve by both our models and stepwise heuristics. One of the purposes of this study is to establish the solution quality of the stepwise heuristic versus the optimal solutions. For these reasons, we generated synthetic instances. Furthermore, we want a large variety of instances with regard to the size and by randomly generating instances, we were also able to achieve this.

For the thin case ($m < n$), we generate 26 sets of instances with $\{(m,n)|m \in \{20,30,40,50\}, n \in \{30,40,\cdots,90,100\}, m+10 \leq n\}$, where each set contains 10 instances. Hence, we generate a total of 260 instances. For the fat case ($m > n$), we generate 16 sets of instances with $\{(m,n)|m \in \{100,150,200,250\}, n \in \{30,40,50,60\}\}$, in which each set contains 10 instances. Hence, we generate a total of 160 instances. For the detailed procedure used to generate the instances, see Section 8 of the online supplement.

To evaluate the performance of the proposed models and algorithms, we compare the improvement against benchmark packages and algorithms. For the thin case with MAE objective and the fat case with both MAE and MSE objectives, we implemented a stepwise algorithm in C#, due to the absence of a statistical package that supports such cases. The algorithm is presented in Section 3 of the online supplement. For the thin case with the MSE objective, we use the stepwise regression implementation of R statistics package Leaps by Lumley [23], which supports the adjusted r^2 objective. The leaps package also provides leaps-and-bound, an exact algorithm proposed by Furnival and Wilson [14]. However, in Section 4 of the online supplement, we show that its complexity is much worse than that of our algorithms. For the remaining portion of the paper, we refer to all of the benchmark algorithms and packages as *Step*. For all proposed models and algorithms, solutions obtained by *Step* are used as initial solutions. As we discussed in the introduction, enumerating all possible subsets is not a computationally tractable approach and it is excluded in the comparison.

For comparison purposes, we use the following measures.

GAP$_{IP}$: the optimality gap obtained by CPLEX within allowed time.
GAP$_{sol}$: relative gap between a proposed model and heuristic defined as

$$\frac{obj \text{ of } Step - obj \text{ of proposed model}}{obj \text{ of } Step}.$$

Solving the problems optimally for larger instances takes a long time as implied in Section 4 of the online supplement. Hence, we set up time limits for CPLEX. We execute CPLEX with two settings for the time limit: one hour and one minute. The computation time of the big M is less than 90 seconds for all instances considered in the experiment, and we do not include this time within the one hour and one minute time limits.

Finally, we summarize the algorithms used for the experiment in Table 1. Recall that we only presented the result for big M with the MAE and MAE_a objectives. For the MSE and MSE_a objectives, we need a trivial modification. In all algorithms and models, to obtain big M for v_j, we use (9) and (31) for the thin and fat cases, respectively. However, we have several options to obtain the big M value for x_j: (12), (32), and procedures in Appendix B. Among these, for the thin case and each iteration of *CoreHeur* and *CoreRnd* for the fat case, we use (12) for big M for x_j, because in each iteration we deal with the thin case. For the fat case MIP models, we use (32) for big M for x_j because other procedures give extremely large values of M. These choices were made based on computational experiments in Section 5 of the online supplement. The result in the online supplement implies that valid big M values guarantee optimality while they do not significantly increase the execution times. Even if CPLEX terminates due to the time limit, the solution qualities are similar regardless of the big M values as long as the big M values are valid.

Case	Obj	Notation	Reference
Thin	MAE	MIP	(6) with big M based on (9) and (12)
Thin	MSE	MIP	(8) with big M based on (9) and (12)
Thin	mRMR	MIP	(19), (19) does not have big M
Fat	MAE_a	MIP	(21) with big M based on (31) and (32)
		CoreHeur	Algorithm 1 with Algorithm 2 and big M based on (9) and (12) with $J := C$
		CoreRnd	Algorithm 1 with Algorithm 3 and big M based on (9) and (12) with $J := C$
Fat	MSE_a	MIP	(36) with big M based on (31) and (32)
		CoreHeur	Algorithm 1 with Algorithm 2 and big M based on (9) and (12) with $J := C$
		CoreRnd	Algorithm 1 with Algorithm 3 and big M based on (9) and (12) with $J := C$

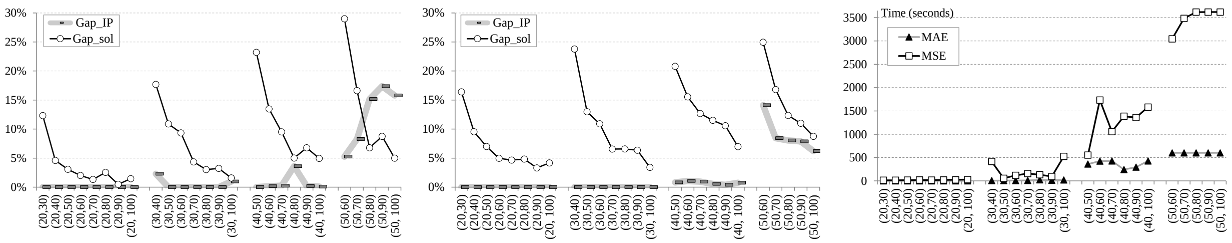
Table 1: Summary of the algorithms

We also note here that big M-based formulations we propose outperform logical constraint-based formulations that are available in CPLEX and most commercial optimization solvers. In Section 6 of the online supplement, we compare the two approaches and observe that the proposed formulations terminate faster with an optimal solution or terminate with a better solution (smaller optimality gap and smaller objective function value) when one minute time limit is employed.

4.2 Study of Thin Case ($m < n$) for MAE and MSE Objectives

In Figure 2, we present the averages of GAP$_{IP}$ and GAP$_{sol}$ across the 26 instance sets. Each rectangle and circle corresponds to the average GAP$_{IP}$ and GAP$_{sol}$ of 10 instances for the corresponding instance set. In both plots on the left, x and y axes represent the instance sets and the gaps in percentage. For both MSE and MAE, GAP$_{IP}$ is near zero for most of the instances with $m \leq 40$. Hence, we get an optimal solution within one hour. For larger instances, GAP$_{IP}$ is positive for both MSE and MAE and is larger for MSE. For GAP$_{sol}$, we observe common phenomena for both objectives. First, GAP$_{sol}$ tends to decrease as n increases for each fixed m. Second, there are bumps for GAP$_{sol}$ at $(m, n) \in \{(20, 30), (30, 40), (40, 50), (50, 60)\}$. Figure 2 also implies that the performance of heuristics deteriorates when we have relatively fewer observations given fixed m, because GAP$_{sol}$ is an underestimation of the gap between an optimal solution and heuristic solution. We also plot the average execution time of (6) and (8). Observe that the average time of (6) for large instances is still 500 seconds, while GAP$_{IP}$ is positive for the same instance sets. This implies that most of the instances are solved optimally but we terminate with a relatively large GAP$_{IP}$ for a few instances after one hour.

During the experiment, we observed that the improvement of the objective function value occurs in the early stage of the branch-and-bound algorithm, and CPLEX tries to improve the lower bound for the remaining time. In Figure 3, we present the primal and lower bounds for one instance over time. The circles and empty circles are the primal and lower bounds over time, respectively, and the plain and dotted lines represent the best primal and lower bounds obtained after one hour. Observe that there is no objective function value improvement after 90 and 25 seconds for MSE and MAE, respectively. In other words, we can obtain the same regression models obtained with one hour execution by terminating CPLEX after 90



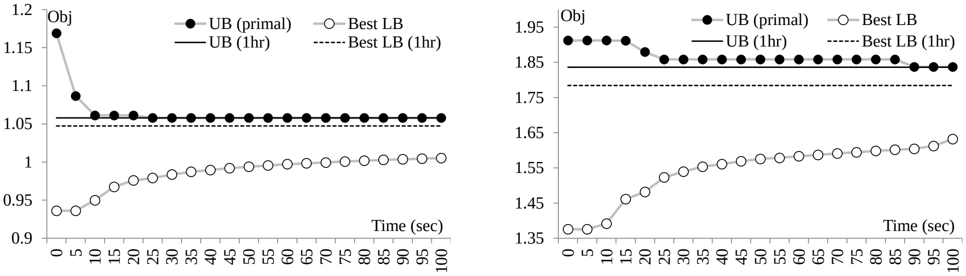
(a) GAP$_{IP}$ and GAP$_{sol}$ for MSE

(b) GAP$_{IP}$ and GAP$_{sol}$ for MAE

(c) MIP Execution time

Figure 2: Average GAP$_{IP}$, GAP$_{sol}$, and execution time with the one hour time limit

seconds. From this observation, we conclude that good solutions are obtained in the early stages of the branch-and-bound algorithm but improving the lower bound takes longer time. This observation gives the justification to run CPLEX for a short time if we do not need to retain optimality.



(a) MSE

(b) MAE

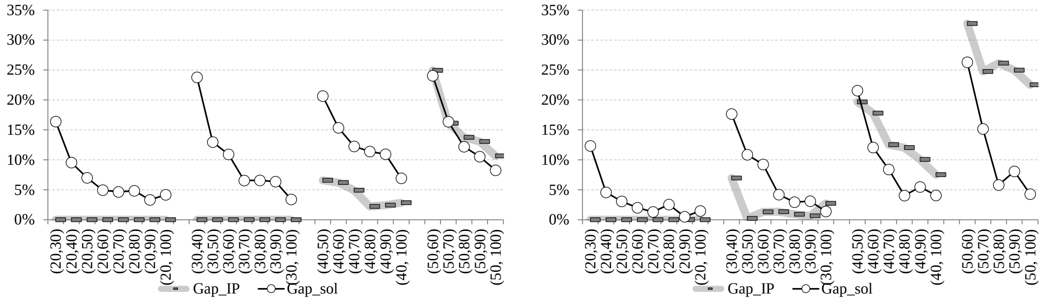
Figure 3: Convergence of primal and dual bounds for an instance with $m = 50$ and $n = 100$

For this reason, we execute CPLEX with the one minute time limit. In the experiment of Bertsimas *et al.* [3], time limit of 500 seconds for MIP is considered as they solve different formulation with larger data. In Figure 4, we present the averages of GAP$_{IP}$ and GAP$_{sol}$ over 26 instance sets, when CPLEX terminates after one minute. We observe a similar shape for GAP$_{sol}$ except the gaps are slightly smaller. On the other hand, GAP$_{IP}$ is positive for more instances compared to the previous result with the one hour time limit. To compare the solution qualities precisely, in Figure 5, we plot the improvement of the primal and lower bounds obtained by executing the extra 59 minutes, where the data points represent $lost(\text{GAP}_{sol}) = \big(\text{GAP}_{sol}$ with one hour - GAP$_{sol}$ with one minute$\big)$ and $lost(\text{GAP}_{IP}) = \big(\text{GAP}_{IP}$ with one minute - GAP$_{IP}$ with one hour$\big)$. Observe that the difference of GAP$_{sol}$ is less than 5% for all cases, whereas there exists significant improvement of the lower bounds for $m \geq 30$. Therefore, within one minute (excluding the big M time), we can improve the stepwise heuristic solution up to 25% by solving the proposed MIP models.

4.3 Study of Thin Case ($m < n$) for Minimal-Redundancy-Maximal-Relevance

In this section, four MIP models are compared: MIP$_{\text{mrmr}}$ (MIP model (18) maximizing mRMR), MIP$_{\text{mae}}$ (MIP model (6)), MIP$_{\text{sae}}$ (MIP model (6) with fixed p minimizing SAE), and MIP$_{\text{mix}}$ (MIP model (19) minimizing SAE subject to the mRMR constraint).

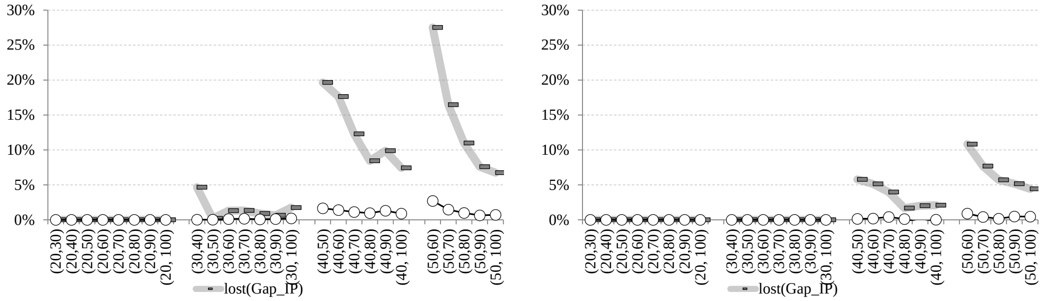
In the first experiment, MIP$_{\text{mix}}$ is compared with MIP$_{\text{mrmr}}$ and MIP$_{\text{sae}}$ for fixed p values. In the second experiment, MIP$_{\text{mix}}$ is compared with MIP$_{\text{mae}}$. Let S_{mrmr}, S_{sae}, and S_{mix} be the selected subsets of the corresponding MIP models, and let mRMR$_{\text{mrmr}}$ and mRMR$_{\text{mixed}}$ be the mRMR values for S_{mrmr} and S_{mix}, respectively. Let SAE$_{\text{sae}}$ and SAE$_{\text{mixed}}$ be the SAE values for S_{sae} and S_{mix}, respectively. To compare the selected subset and solution quality of MIP$_{\text{mix}}$ against the other three models, the following criteria are used. For each $model \in \{mrmr, mae, sae\}$, set difference between S_{model} and S_{mixed},



(a) MSE

(b) MAE

Figure 4: Average GAP_{IP} and GAP_{sol} with the one minute time limit



(a) MSE

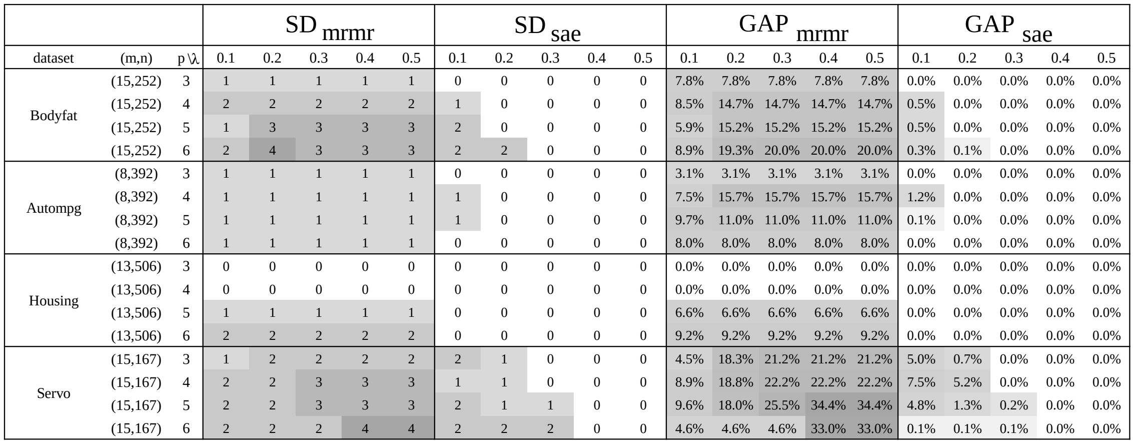
(b) MAE

Figure 5: Average improvement of GAP_{IP} and GAP_{sol} by the extra 59 minutes

$\text{SD}_{\text{model}} = \frac{|(S_{\text{model}} \setminus S_{\text{mix}})| + |(S_{\text{mix}} \setminus S_{\text{model}})|}{2}$, is defined. For all four models, the relative mRMR gap from MIP_{mrmr} ($\text{GAP}_{\text{mrmr}}(\%) = \frac{\text{mRMR}_{\text{mrmr}} - \text{mRMR}_{\text{model}}}{\text{mRMR}_{\text{mrmr}}} \times 100$) and relative SAE gap from the optimal SAE ($\text{GAP}_{\text{sae}}(\%) = \frac{\text{SAE}_{\text{model}} - \text{SAE}_{\text{sae}}}{\text{SAE}_{\text{sae}}} \times 100$) are defined. Note that SD_{mrmr} and SD_{sae} measure how the selected subset by MIP_{mix} is different from the subsets obtained by MIP_{mrmr} and MIP_{sae}, respectively. To measure the solution quality in terms of mRMR and SAE, GAP_{mrmr} and GAP_{sae} calculate the relative gaps of MIP_{mix} from the best mRMR (by MIP_{mrmr}) and best SAE (by MIP_{sae}), respectively.

To test the performances of the models with various parameters and sizes, we conduct experiments using the thin case synthetic data from Section 4.2 and report the result in Section 9 of the online supplement. The result of these experiments confirms that MIP_{mix} effectively balances the mRMR and SAE objects. The obtained subset by MIP_{mix} is distinguished from the subsets of MIP_{mrmr} and MIP_{sae}. Check the online supplement for the detailed results. For the experiments in this section, the MIP models are tested using select real datasets from the UCI Machine Learning Repository [22] and Johnson [31]. Four regression datasets (Bodyfat, Autompg, Housing, and Servo) are selected among the datasets with more than 100 observations and that are created for linear regression analysis. The original data are processed by deleting rows with missing values and by creating dummy variables for categorical variables. All final variables are standardized.

In the first experiment, for each dataset, parameters $p \in \{3, 4, 5, 6\}$ and $\lambda \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ are used. In Figure 6, a heatmap is presented for the four performance measures SD_{mrmr}, SD_{sae}, GAP_{mrmr}, and GAP_{sae}. The execution times are not reported because all models are solved optimally within a second. The rows are defined for datasets and p, and the columns are defined for λ values. The heatmap shows the same trend with the previous experiments. Increasing λ and p values increases SD_{mrmr} and GAP_{mrmr} while decreases SD_{sae} and GAP_{sae}. For several cases (Housing data with $p = 3, 4, 5$), $\text{SD}_{\text{mrmr}} = \text{SD}_{\text{sae}} = 0$ because the selected subset is optimal for both criteria mRMR and SAE. For several cases ($\lambda = 0.4, 0.5$ for Augompg, Housing, Servo), $\text{SD}_{\text{sae}} = 0$ because S_{sae} has the mRMR value within 40% from the optimal mRMR value, which also implies Constraint (19d) does not cut any part of the feasible region. In order to determine the best balance between the two criteria, a user can determine an allowable maximum for any of the gaps GAP_{mrmr} and GAP_{sae} and select the best in the scope. Otherwise, a pareto frontier and scatter plot can be useful in selecting a good solution.



			SD mrmr					SD sae					GAP mrmr					GAP sae				
dataset	(m,n)	p \λ	0.1	0.2	0.3	0.4	0.5	0.1	0.2	0.3	0.4	0.5	0.1	0.2	0.3	0.4	0.5	0.1	0.2	0.3	0.4	0.5
Bodyfat	(15,252)	3	1	1	1	1	1	0	0	0	0	0	7.8%	7.8%	7.8%	7.8%	7.8%	0.0%	0.0%	0.0%	0.0%	0.0%
	(15,252)	4	2	2	2	2	2	1	0	0	0	0	8.5%	14.7%	14.7%	14.7%	14.7%	0.5%	0.0%	0.0%	0.0%	0.0%
	(15,252)	5	1	3	3	3	3	2	0	0	0	0	5.9%	15.2%	15.2%	15.2%	15.2%	0.5%	0.0%	0.0%	0.0%	0.0%
	(15,252)	6	2	4	3	3	3	2	2	0	0	0	8.9%	19.3%	20.0%	20.0%	20.0%	0.3%	0.1%	0.0%	0.0%	0.0%
Autompg	(8,392)	3	1	1	1	1	1	0	0	0	0	0	3.1%	3.1%	3.1%	3.1%	3.1%	0.0%	0.0%	0.0%	0.0%	0.0%
	(8,392)	4	1	1	1	1	1	1	0	0	0	0	7.5%	15.7%	15.7%	15.7%	15.7%	1.2%	0.0%	0.0%	0.0%	0.0%
	(8,392)	5	1	1	1	1	1	1	0	0	0	0	9.7%	11.0%	11.0%	11.0%	11.0%	0.1%	0.0%	0.0%	0.0%	0.0%
	(8,392)	6	1	1	1	1	1	0	0	0	0	0	8.0%	8.0%	8.0%	8.0%	8.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Housing	(13,506)	3	0	0	0	0	0	0	0	0	0	0	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	(13,506)	4	0	0	0	0	0	0	0	0	0	0	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	(13,506)	5	1	1	1	1	1	0	0	0	0	0	6.6%	6.6%	6.6%	6.6%	6.6%	0.0%	0.0%	0.0%	0.0%	0.0%
	(13,506)	6	2	2	2	2	2	0	0	0	0	0	9.2%	9.2%	9.2%	9.2%	9.2%	0.0%	0.0%	0.0%	0.0%	0.0%
Servo	(15,167)	3	1	2	2	2	2	2	1	0	0	0	4.5%	18.3%	21.2%	21.2%	21.2%	5.0%	0.7%	0.0%	0.0%	0.0%
	(15,167)	4	2	2	3	3	3	1	1	0	0	0	8.9%	18.8%	22.2%	22.2%	22.2%	7.5%	5.2%	0.0%	0.0%	0.0%
	(15,167)	5	2	2	3	3	3	2	1	1	0	0	9.6%	18.0%	25.5%	34.4%	34.4%	4.8%	1.3%	0.2%	0.0%	0.0%
	(15,167)	6	2	2	2	4	4	2	2	2	0	0	4.6%	4.6%	4.6%	33.0%	33.0%	0.1%	0.1%	0.1%	0.0%	0.0%

Figure 6: Performances of MIP_{mix} compared to MIP_{mrmr} and MIP_{sae}

In the second experiment, MIP_{mix} is compared with our MIP model (6), which we denote as MIP_{mae}. While MIP_{sae} assumes fixed p, our MIP model (7) from Section 2.1 can be used to find the optimal p value, referred to as p^*. Hence, we solved (7) to obtain p^* and the optimal MAE. Then, we compare the solution quality of MIP_{mix} by fixing p to p^* and by checking various λ values. In Table 2, the fourth column represents the relative gap of the mRMR objective between (7) and optimal mRMR, the fifth column represents the relative gap of the MAE objective between (7) and MIP_{mix}, the sixth column represents the relative gap of the mRMR objective between MIP_{mix} and optimal mRMR, and the last column represents the set difference between (7) and MIP_{mix}.

			MIP_{mae}		MIP_{mix}		
Dataset	(m,n)	λ	p^*	GAP_{mrmr}	GAP_{mae}	GAP_{mrmr}	SD_{mae}
Bodyfat	(15,252)	0.05	4	14.7%	0.6%	4.7%	2
		0.1			0.5%	8.5%	1
		0.15			0.0%	14.7%	0
		0.2			0.0%	14.7%	0
Autompg	(8,392)	0.05	4	15.7%	1.2%	1.0%	1
		0.1			1.2%	7.5%	1
		0.15			1.2%	7.5%	1
		0.2			0.0%	15.7%	0
Housing	(13,506)	0.05	11	6.5%	1.3%	4.7%	2
		0.1			0.0%	6.5%	0
		0.15			0.0%	6.5%	0
		0.2			0.0%	6.5%	0
Servo	(15,167)	0.05	9	8.1%	1.1%	4.8%	1
		0.1			0.0%	8.1%	0
		0.15			0.0%	8.1%	0
		0.2			0.0%	8.1%	0

Table 2: Comparison with MAE model

The GAP_{mrmr} values of MIP_{mae} show that the optimal MAE subset is quite different from the optimal mRMR subset and the mRMR values are different up to 14.7%. By MIP_{mix}, we can improve the mRMR value significantly without decreasing MAE too much. For all four datasets, with $\lambda = 0.05$, GAP_{mrmr} values of MIP_{mix} are significantly lower than those of MIP_{mae}, while GAP_{mae} values of MIP_{mix} are approximately 1% from the optimal MAE value. In detail, for Autompg data, MIP_{mix} keeps both of GAP_{mae} and GAP_{mrmr} approximately at 1%.

4.4 Study of Fat Case ($m > n$)

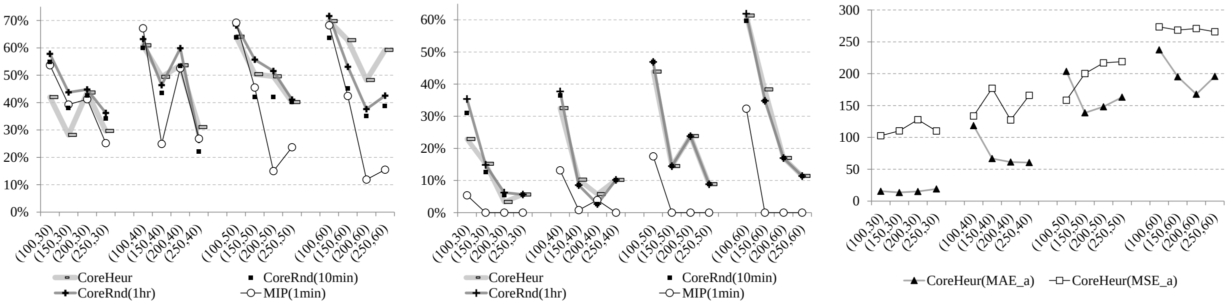
In this section, we present two experiments for the fat case datasets. In the first experiment, the solution qualities of the MIP models, (21) and (36), and the core set algorithms, *Core-Heuristic* and *Core-Random*, are compared using the synthetic datasets. In the second experiment, the core set algorithms are compared against the stepwise algorithm and a state-of-the-art benchmark algorithm using real-world instances from the UCI Machine Learning Repository.

Recall that the core set algorithms require core set cardinality parameter θ. Hence, we first decide the best θ value for each core set algorithm, then we compare *Core-Heuristic*, *Core-Random*, and the MIP models in Section 7 of the online supplement. We conclude the following universal rule for the selection of θ.

1. For *Core-Heuristic*, we use $\theta = 1$ for instance sets satisfying $\{\frac{n}{m} \geq 0.4, n \leq 40\}$ or $\{\frac{n}{m} \geq 0.5, n > 40\}$. For all other instances, we use $\theta = 0.8$.
2. For *Core-Random*, with a ten minute time limit, $\theta = 1.0$ is best for all sizes.
3. For *Core-Random*, with a one hour time limit, $\theta = 0.8$ is best for large instances. Hence, with the one hour time limit, we use $\theta = 0.8$ if $mn \geq 9000$ and $\theta = 1.0$ otherwise.

We compare GAP_{sol} of the MIP models, and *Core-Heuristic* and *Core-Random* with the best θ determined by the rule above. In Figure 5, we observed that running the MIP solver beyond 1 minute does not improve the solution quality much. For this reason, to save computational power, we ran the MIP solver for 1 minute for the fat case. For *Core-Random*, we set 10 minutes and 1 hour time limit to check the performance as we spend more time.

For the first experiment, we present the average GAP_{sol} for all algorithms and execution times for *Core-Heuristic* in Figure 7. For the MSE_a objective, MIP performs worst for all instances. For many instance sets, it does not improve the initial heuristic solution. *Core-Random* performs slightly better than *Core-Heuristic* for small instances with $n = 30$, but they perform equally for remaining instances. For the MAE_a objective, the performance of MIP drops substantially when m increases. For most instances, *Core-Random* performs the best in general. However, for larger instances with $n = 60$, *Core-Heuristic* performs the best.



(a) GAP_{sol} for MSE_a (b) GAP_{sol} for MAE_a (c) *Core-Heuristic* execution time

Figure 7: Comparison of performance of the algorithms

For the second experiment, we compare the performance of the core set algorithms with two benchmark algorithms: a stepwise heuristic minimizing MSE and the mathematical programming based algorithm of Bertsimas *et al.* [3]. We use the R package *bestsubset* of Hastie et al. [18] which implements Bertsimas *et al.* [3]. We denote this algorithm as *BKM*. For this experiment, we use two sets of the dataset of Rafiei and Adeli [29] from the UCI Machine Learning Repository [22] that have more than 100 features and that are created for linear regression analysis. The original dataset has 103 features and two possible response variables cost and sales. To create fat case datasets, we randomly select 50 observations and create 10 instances for each response variable. All explanatory variables are standardized.

We use a ten minute time limit for BKM to compare with the core set algorithms. Note that, within this time limit, BKM may not guarantee optimality and that BKM requires a fixed p. To search for the best MSE and within the ten minute time limit, we enumerate BKM with the following search order for p:

1,3,5,7, $\cdots$,45,47,2,4,6,$\cdots$,46,48. For each p, 60 seconds is allowed and the algorithm stops at 600 seconds even if all p values were not searched.

		Gap from the best				Time (seconds)			
data	(n,m)	BKM (10min)	Step	CoreHeur	CoreRnd (10min)	BKM (10min)	Step	CoreHeur	CoreRnd (10min)
Cost1	(50,103)	33.5%	105.2%	12.9%	**0.0%**	600	10	77	648
Cost2	(50,103)	53.8%	6.9%	**0.0%**	**0.0%**	600	12	98	646
Cost3	(50,103)	5.4%	23.2%	8.5%	**0.0%**	600	10	141	632
Cost4	(50,103)	5.4%	17.2%	**0.0%**	8.9%	600	9	204	634
Cost5	(50,103)	23.9%	14.4%	**0.0%**	3.0%	600	11	141	647
Cost6	(50,103)	16.5%	50.1%	**0.0%**	5.8%	600	4	134	624
Cost7	(50,103)	**0.0%**	30.8%	26.7%	3.1%	600	8	136	634
Cost8	(50,103)	97.2%	18.7%	13.0%	**0.0%**	600	11	52	581
Cost9	(50,103)	7.3%	11.2%	3.8%	**0.0%**	600	7	137	633
Cost10	(50,103)	19.6%	10.4%	4.8%	0.0%	602	11	142	650
Sales1	(50,103)	10.7%	28.3%	**0.0%**	1.6%	602	6	138	628
Sales2	(50,103)	73.8%	363.0%	2.0%	**0.0%**	601	8	148	668
Sales3	(50,103)	**0.0%**	44.1%	44.0%	44.0%	600	6	138	638
Sales4	(50,103)	1.3%	32.4%	2.3%	**0.0%**	600	5	135	642
Sales5	(50,103)	7.8%	4.6%	4.6%	**0.0%**	600	12	141	652
Sales6	(50,103)	**0.0%**	29.9%	29.9%	4.2%	600	8	139	645
Sales7	(50,103)	**0.0%**	50.1%	37.7%	11.4%	600	8	139	640
Sales8	(50,103)	21.2%	**0.0%**	**0.0%**	**0.0%**	601	12	55	650
Sales9	(50,103)	**0.0%**	31.5%	6.9%	13.2%	602	6	202	629
Sales10	(50,103)	**0.0%**	5.5%	5.3%	4.2%	600	8	137	628
	Average	18.9%	43.9%	10.1%	5.0%	600	9	132	637

Table 3: Performance of core set and benchmark algorithms with ten minutes time limit

The result for the second experiment is presented in Table 3. The first two columns describe the datasets, the next four columns present the gap of each algorithm from the best objective value of the four algorithms, and the last four columns report the running time. The smallest gap among the four gaps is in boldface. The stepwise algorithm is the fastest while the gap from the best algorithm is over 40% on average. The three MIP-based algorithms do not dominate each other: BKM wins six cases, CoreHeur wins six cases, and CoreRnd wins 9 cases. However, the relative gap of CoreRnd is the smallest, which show the effectiveness and robustness of the algorithm given the ten minute time limit. CoreHeur can be a good alternative to CoreRnd because it spends significantly less time than the other two MIP-based algorithms and quickly improves the solution quality of the stepwise algorithm.

5 Conclusion

In this study, we present mathematical programs to optimize various subset selection criteria: MAE, MSE, mRMR, and variants. The proposed mathematical programs return an optimal subset given a valid value of big M, which is also derived in our work. For the selected test instances, we observe that the solver frequently spends more than an hour to prove optimality, while near-optimal solutions are obtained in the first minute. To speed up the solution time and to deal with high dimensional cases, we propose an iterative algorithm based on the popular core set concept. The proposed algorithm and the randomized version converge to local and global optimal solutions, respectively, and show that they outperform the state-of-the-art benchmark.

Mathematical programming models for subset selection are getting rapidly increasing attention recently due to the improved computational power and numerical solver efficiency. Further, the use of binary decision variables can help to model various subset requirements such as conditional inclusion (exclusion) of explanatory variables. Despite the benefits, there are still limitations in the current mathematical programming models. For example, the current approaches cannot solve large scale instances (e.g., millions of observations or explanatory variables) optimally. Hence, developing an improved model or an efficient algorithm with guaranteed optimality is crucial. Also, the big M values derived in the current work are valid, but not the tightest; this slows down the branch and bound algorithm speed. Hence, tighter big M values can be further studied.

Acknowledgments

The authors appreciate the editors and reviewers for their constructive comments and suggestions that strengthened the paper.

References

[1] Bertsimas, D. and Weismantel, R. *Optimization Over Integers.* Dynamic Ideas, 2005.

[2] Bertsimas, D. and Shioda, R.(2009). Algorithm for cardinality-constrained quadratic optimization. *Computational Optimization and Applications*, **43**, 1–22.

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

[4] Bertsimas, D. and King, A.(2016). OR forum - an algorithmic approach to linear regression. Operations Research, **64(1)**, 2–16.

[5] Bienstock, D.(1996). Computational study of a family of mixed-integer quadratic programming problems. *Mathematical Programming*, **74**, 121–140.

[6] Bradley, P.S., Mangasarian, O.L., and Street, W.N.(1998). Feature selection via mathematical programming. INFORMS Journal on Computing, **10:2**, 209–217.

[7] Candes, E. and Tao, T.(2007). The Danzig selector: statistical estimation when p is much larger than n. *The Annals of Statistics*, **35**, 2313–2351.

[8] Chai, T. and Draxler, R.R.(2004). Root mean square error (RMSE) or mean absolute error (MAE)? - Arguments against avoiding RMSE in the literature. *Geoscientific Model Development*, **7:3**, 1247–1250.

[9] Charnes, A, Cooper, W.W., and Ferguson, R.O.(1955). Optimal estimation of executive compensation by linear programming. *Management Science*, **1**, 138–151.

[10] de Farias Jr., I.R. and Nemhauser, G.L.(2003). A polyhedral study of the cardinality constrained knapsack problem. *Mathematical Programming*, **96:3**, 439–467.

[11] Dielman, Terry E.(2005). Least absolute value regression: recent contributions. *Journal of Statistical Computation and Simulation*, **75:4**, 263–286.

[12] Ding, C. and Peng, H.(2005). Minimum redundancy feature selection from microarray gene expression data. *Journal of Bioinformatics and Computational Biology*, **3:2**, 185–205.

[13] Fung, G.N. and Mangasarian, O.L.(2004). A feature selection Newton method for support vector machine classification. *Computational Optimization and Applications*, **28:2**, 185–202.

[14] Furnival, G.M. and Wilson, R.W.(1974). Regressions by leaps and bounds. *Technometrics*, **16**, 499–511.

[15] Glover, F. (1975). Improved linear integer programming formulations of nonlinear integer problems. *Management Science*, **22:4**, 455-460.

[16] Har-Peled, S. *Geometric Approximation Algorithms.* American Mathematical Society, 2011.

[17] Harrell, F.E.. *Regression Modeling Strategies: With Applications to Linear Models, Logistic and Ordinal Regression, and Survival Analysis.* Springer, 2001.

[18] Hastie, T., Tibshirani, R., and Tibshirani, R.(2017). Tools for best subset selection in regression. Package "bestsubset."

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

[20] Hwang, K., Kim, D., Lee, K., Lee, C. and Park, S.(2017). Embedded variable selection method using signomial classification. *Annals of Operations Research*, **254**, 89–109.

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

[22] Lichman, M.(2013). UCI machine learning repository. http://archive.ics.uci.edu/ml

[23] Lumley, T.(2009). Leaps: regression subset selection. R package version 2.9. http://CRAN.R-project.org/package=leaps

[24] Miller, A.J. (1984). Selection of subsets of regression variables. *Journal of the Royal Statistical Society. Series A*, **147**, 389–425.

[25] Miller, A.J.. *Subset Selection in Regression.* Chapman and Hall, 2002.

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

[27] Narula, S.C. and Wellington, J.F. (1982). The minimum sum of absolute error regression: a state of the art survey. *International Statistical Review*, **50**, 317–326.

[28] Peng, H., Long, F., and Ding, C. (2005). Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. IEEE Transactions on Pattern Analysis and Machine Intelligence, **27:8**, 1226–1238.

[29] Rafiei, M.H. and Adeli, H. (2015). A novel machine learning model for estimation of sale prices of real estate units. Journal of Construction Engineering and Management, **142:2**, 04015066.

[30] Rinaldi, F. and Sciandrone, M. (2010). Feature selection combining linear support vector machines and concave optimization. Optimization Methods and Software, **25:1**, 117–128.

[31] Johnson, R. W. (1996). Fitting percentage of body fat to simple body measurements. Journal of Statistics Education, **4:1**.

[32] Schaible, S. and Shi, J. (2004). Recent developments in fractional programming: single-ratio and max-min case. Nonlinear Analysis and Convex Analysis, 493-506.

[33] Stancu-Minasian, IM (2012). *Fractional programming: theory, methods and applications.* Springer Science & Business Media.

[34] Schlossmacher, E.J. (1973). An iterative technique for absolute deviations curve fitting. *Journal of the American Statistical Association*, **68**, 857–859.

[35] Schrijver, A (1998). *Theory of linear and integer programming.* Jone Wiley & Sons.

[36] Stodden, V. (2006) Model selection when the number of variables exceeds the number of observations. PhD dissertation. Stanford University.

[37] Tamhane, A.C. and Dunlop, D.D.. *Statistics and Data Analysis: From Elementary to Intermediate.* Pearson, 1999.

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

[39] Wagner, H.M. (1959) Linear programming techniques for regression analysis. *Journal of the American Statistical Association*, **54**, 206–212.

[40] Western, J., Elisseeff, A., Schölkopf, B. and Tipping, M. (2003). Use of the zero-norm with linear models and kernel methods. Journal of Machine Learning Research, **3**, 1439–1461.

[41] Willmott, C.J. and Matsuura, K.. (2005) Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate Research*, **30:1**, 79–82.

APPENDIX

A Proof of Lemmas and Propositions

Proof of Proposition 1

The proof is based on the fact that feasible solutions to (4) and (5) map to each other. Hence, we consider the following two cases.

1. Case: (4) ⇒ (5)
   Let $S = \{j | z_j = 1\}$ be the column index set of a solution to (4). We set $v_j = u$ for $j \in S$ and $v_j = 0$ for $j \notin S$. Then,

   $$\begin{aligned} \textstyle\sum_{i \in I} |t_i| &= (n-1)u - \textstyle\sum_{j \in J} u z_j && \text{(from (4b))} \\ &= (n-1)u - \textstyle\sum_{j \in S} u \\ &= (n-1)u - \textstyle\sum_{j \in S} v_j && \text{(by definition of } v_j\text{)} \\ &= (n-1)u - \textstyle\sum_{j \in J} v_j, \end{aligned}$$

   which satisfies (5). Further, we satisfy the following.

   (a) Constraint (5e): We have $v_j = u \leq u$ for $j \in S$ and $v_j = 0 \leq u$ for $j \notin S$. Hence, $v_j \leq u$ for all $j \in J$.

   (b) Constraint (5f): We have $u - M(1 - z_j) = u \leq v_j = u \leq M z_j = M$ for $j \in S$ and $u - M(1 - z_j) = u - M \leq v_j = 0 \leq M z_j = 0$ for $j \notin S$. Hence, we satisfy (5f).

   (c) Constraint (5g): We have $v_j \in \{0, u\} \geq 0$, for all $j \in J$.

   Note that (5c) is automatically satisfied since it is equal to (4c). Hence, we obtain a feasible solution to (5).

2. Case: (5) ⇒ (4)
   Let $S = \{j | z_j = 1\}$ be the column index set of a solution to (5). Since we are minimizing u, (5e) is equivalent to $\max_j v_j = u$. Note that, in an optimal solution, we must have $v_j = u$ for all $j \in S$. Hence, starting from (5b), we derive

   $$\begin{aligned} \textstyle\sum_{i \in I} |t_i| &= (n-1)u - \textstyle\sum_{j \in J} v_j && \text{(from (5b))} \\ &= (n-1)u - \textstyle\sum_{j \in S} v_j = (n-1)u - \textstyle\sum_{j \in S} u && (v_j = u \text{ for all } j \in S) \\ &= (n-1)u - \textstyle\sum_{j \in S} u z_j = (n-1)u - \textstyle\sum_{j \in J} u z_j, \end{aligned}$$

   which satisfies (5).

This ends the proof. □

Proof of Proposition 2

Let $\bar{X} = (\bar{x}, \bar{y}, \bar{v}, \bar{u}, \bar{t}, \bar{z})$ be an optimal solution to (6) and let $\bar{p} = \sum_{j \in J} \bar{z}_j$ be the number of optimal regression variables. For a contradiction, let us assume that there exists an index k such that $\bar{t}_k^+ > 0$ and $\bar{t}_k^- > 0$. Without loss of generality, let us also assume $\bar{t}_k^+ \geq \bar{t}_k^-$. For simplicity, let $\delta = \bar{t}_k^-$. Let us generate $\tilde{X}$ that is equal to $\bar{X}$ except $\tilde{t}_k^+ = \bar{t}_k^+ - \delta$, $\tilde{t}_k^- = \bar{t}_k^- - \delta = 0$, $\tilde{u} = \bar{u} - \frac{2\delta}{n-1-\bar{p}}$, and $\tilde{v}_j = \tilde{u}$ if $\bar{z}_j = 1$. We show that $\tilde{X}$ is a feasible solution to (6) with strictly lower cost than $\bar{X}$.

1. $\tilde{X}$ has lower cost than $\bar{X}$ since $\tilde{u} < \bar{u}$ by definition.
2. $\tilde{X}$ satisfies (6b) because $\sum_{i \in I}(\tilde{t}_i^+ + \tilde{t}_i^-) = \sum_{i \in I}(\bar{t}_i^+ + \bar{t}_i^-) - 2\delta = (n-1)\bar{u} - \sum_{j \in J} \bar{v}_j - 2\delta = (n - 1 - \bar{p})\bar{u} - 2\delta = (n - 1 - \bar{p})(\bar{u} - \frac{2\delta}{n-1-\bar{p}}) = (n - 1 - \bar{p})\tilde{u} = (n-1)\tilde{u} - \sum_{j \in J} \tilde{v}_j$, in which the second equality holds because $\bar{X}$ satisfies (6b).
3. Observe that (6c), (6d), and (6e) are automatically satisfied. Further, since we set $\tilde{v}_j = \tilde{u}$ for j such that $\tilde{z}_j = 1$, (6f) and (6g) are satisfied.

4. Finally, (6h) is automatically satisfied except for $\tilde{t}_k^+$,$\tilde{t}_k^-$, and $\tilde{u}$. Note that $\tilde{t}_k^+ = \bar{t}_k^+ - \delta = \bar{t}_k^+ - \bar{t}_k^- \geq 0$ and $\tilde{t}_k^- = 0$. Also, we have

$$\begin{aligned}
\tilde{u} &= \bar{u} - \tfrac{2\delta}{n-1-\bar{p}} = \tfrac{\sum_{i\in I}(\tilde{t}_i^+ + \tilde{t}_i^-)}{n-1-\bar{p}} - \tfrac{2\delta}{n-1-\bar{p}} \\
&= \tfrac{\sum_{i\in I\setminus\{k\}}(\tilde{t}_i^+ + \tilde{t}_i^-) + (\tilde{t}_k^+ + \tilde{t}_k^-) - 2\delta}{n-1-\bar{p}} \\
&\geq \tfrac{\sum_{i\in I\setminus\{k\}}(\tilde{t}_i^+ + \tilde{t}_i^-) + 2\tilde{t}_k^- - 2\delta}{n-1-\bar{p}} && \text{(since } \tilde{t}_k^+ \geq \tilde{t}_k^-\text{)} \\
&= \tfrac{\sum_{i\in I\setminus\{k\}}(\tilde{t}_i^+ + \tilde{t}_i^-)}{n-1-\bar{p}} && \text{(by the definition of } \delta\text{)} \\
&\geq 0.
\end{aligned}$$

Hence, $\tilde{X}$ satisfies (6h).

Hence, $\bar{X}$ is not an optimal solution to (6), which is a contradiction. □

Proof of Proposition 3

Let $\bar{X} = (\bar{x}, \bar{y}, \bar{v}, \bar{u}, \bar{t}, \bar{z})$ be an optimal solution to (8) with $\bar{p} = \sum_{j\in J} \bar{z}_j$. For a contradiction, let us assume that $\bar{X}$ does not satisfy (7) at equality. Let $\delta = (n-1)\bar{u} - \sum_{j\in J} \bar{v}_j - \sum_{i\in I}(\bar{t}_i^+ - \bar{t}_i^-)^2 > 0$. Let us generate $\tilde{X}$ that is equivalent to $\bar{X}$ except that $\tilde{u} = \bar{u} - \frac{2\delta}{n-1-\bar{p}}$ and $\tilde{v}_j = \tilde{u}$ if $\bar{z}_j = 1$. We first show that $\tilde{u} \geq 0$ since

$$\begin{aligned}
\tilde{u} &= \tfrac{\bar{u}(n-1-\bar{p}) - 2\delta}{n-1-\bar{p}} = \tfrac{\bar{u}(n-1) - \bar{u}\bar{p} - 2(n-1)\bar{u} + 2\sum_{j\in J}\bar{v}_j + 2\sum_{i\in I}(\bar{t}_i^+ - \bar{t}_i^-)^2}{n-1-\bar{p}} \\
&= \tfrac{\sum_{j\in J}\bar{v}_j - \bar{u}(n-1) + +2\sum_{i\in I}(\bar{t}_i^+ - \bar{t}_i^-)^2}{n-1-\bar{p}} = \tfrac{\delta}{n-1-\bar{p}} + \tfrac{\sum_{i\in I}(\bar{t}_i^+ - \bar{t}_i^-)^2}{n-1-\bar{p}} \geq \tfrac{\delta}{n-1-\bar{p}} \geq 0,
\end{aligned}$$

in which the second equality is obtained by the definition of δ. For the remaining part, using a similar technique as in the proof of Proposition 2, it can be seen that $\tilde{X}$ is a feasible solution to (8) with strictly lower objective function value than $\bar{X}$. This is a contradiction. □

Lemma 6. Let c be a vector that has 1 for t_i^+'s and t_i^-'s and 0 for all other variables of (10). Then, for every extreme ray r in the recession cone of (10), we must have $c^\top r > 0$.

Proof. Suppose that there exists extreme ray r in the recession cone of (10) with $c^\top r \leq 0$. Let us consider linear program min $\{c^\top Y |$ (10a) - (10e) $\}$. We have two cases.

1. Suppose that $c^\top r < 0$. Note that $\bar{Y} + \delta r$ is feasible for any $\delta \geq 0$ and a feasible solution $\bar{Y}$, since r is extreme ray. Then, $c^\top(\bar{Y} + \delta r) = c^\top \bar{Y} + \delta c^\top r$ goes to negative infinity and thus the LP is unbounded from below. However, from the definition of the LP, the objective value is always non-negative. This is a contradiction.

2. Suppose that $c^\top r = 0$. This implies that the LP has the optimal objective value of 0. This contradicts Assumption 1 since $c^\top Y = 0$ implies $\sum_{i=1}^n (t_i^+ + t_i^-) = 0$.

By the above two cases, we must have $c^\top r > 0$. □

Proof of Proposition 4

From Lemma 6, we know that there is no extreme rays with non-positive $\sum_{i=1}^n (t_i^+ + t_i^-)$. For the proof of the proposition, let us assume that (11) is unbounded and thus there is an extreme ray r such that $\bar{c}^\top r < 0$, where $\bar{c}$ is the objective vector of objective function of (11). Given such extreme ray r, we must have $c^\top r > 0$ by Lemma 6, where c is a vector that has 1 for t_i^*'s and t_i^-'s and 0 for all other variables of (10). For a feasible solution $\bar{Y}$ to (11) and any $\delta \geq 0$, $\bar{Y} = Y + \delta r$ is also feasible. Note that δ must go to infinity for (11) to be an unbounded LP. However, $\delta c^\top r > 0$ implies $\sum_{i\in I}(t_i^+ + t_i^-)$ increases as δ increases. Hence, δ must be bounded by (10a). This implies that $\bar{Y}$ cannot be bounded for any δ. □

Proof of Lemma 1

With fixed $\bar{z}_j$, we have fixed $\bar{v}_j$ and $\bar{u}$ from (6f). Note that, since $\bar{Y}$ has SSE less than or equal to T_{max}, we have $(n-1)\bar{u} - \sum_{j\in J}\bar{v}_j = \sum_{i\in I}(t_i^+ + t_i^-) \leq T_{max}$, which satisfies (10a). Observe that v_j's and u can be ignored in (10). Observe also that (10c) and (10d) cover (6d) and (6e) regardless of $\bar{z}_j$. Finally, (6c) and (10b) are the same. Therefore, $\tilde{Y} = (\bar{x}^+, \bar{x}^-, \bar{y}^+, \bar{y}^-, \bar{t}^+, \bar{t}^-, \hat{M})$ is feasible for (10). □

B Alternative Approach for Big M

In this section, we derive an approximated value for Big M for x_j's in (21) and (36).

Algorithm 4 Estimate-M

```
1: For $k \in J$
2:     For $s = 1, \cdots, 30$
3:         Pick explanatory variable $k$ and $n-3$ explanatory variables randomly and generate new instances with the selected $n-2$ columns and $n$ observations
4:         Solve (1) and set $M_k^s \leftarrow x_k^*$
5:     End-For
6:     $\bar{M}_k \leftarrow average(M_k^1, \cdots, M_k^{30})$, $\sigma^{M_k} \leftarrow std\text{-}dev(M_k^1, \cdots, M_k^{30})$, $\hat{M}_k \leftarrow \bar{M}_k + 1.65\sigma_{M_k}$
7: End-For
```

Instead of trying to get a valid value of M, we use a statistical approach to get an approximated value of M for x_j. In Algorithm 4, we estimate a valid value of M for each k. In Steps 2-5, we obtain 30 i.i.d. sample values of M when explanatory variable k is included in the regression model. Then, in Step 6, we obtain the upper tail of the confidence interval. With 95% confidence, the true valid value of M is less than $\hat{M}$ in Step 6. Hence, we set $M_k := \hat{M}_k$ for x_k in (21) and (36) for the fat case ($m > n$).

C New Objective Function and Modified Formulations for Fat Case ($m \geq n$)

Before we derive the objective function, let us temporarily assume $|J| = n-2$ so that any subset S of J automatically satisfies $|S| = p \leq n-2 = |J|$. We will relax this assumption later to consider $|J| > n-2$. Suppose that we want to penalize large p in a way that the best model with $n-2$ explanatory variables is as bad as a regression model with no explanatory variables. Hence, we want the objective function to give the same value for models with $p = 0$ and $p = n-2$. With this in mind, we propose (20), which we call the adjusted MAE .

Let us now assume that SAE is near zero when $p = n-2$, which happens often. Then we have $MAE_a = \frac{SAE + \frac{n-2}{n-2} mae_0}{n-1-(n-2)} = SAE + mae_0 \approx mae_0$. Hence, instead of near-zero MAE, the new objective has almost the same value as mae_0 when $p = n-2$. Recall that $u = MAE$ and u is the objective function in the previous thin case model. Hence, we need to modify the definitions and constraints. First we rewrite constraint (6b) as $\sum_{i \in I}(t_i^+ + t_i^-) = (n-1)u - \sum_{j \in J} z_j \Big(u + \frac{mae_0}{n-2}\Big)$. Let $v_j = (u + \frac{mae_0}{n-2}) z_j$. Then, (6f) and (6g) are modified to

$$v_j \leq u + \frac{mae_0}{n-2} \tag{28}$$

$$u + \frac{mae_0}{n-2} - M(1 - z_j) \leq v_j \leq M z_j. \tag{29}$$

Finally, we remove the assumption we made ($|J| = n-2$) at the beginning of this section by adding cardinality constraint

$$\textstyle\sum_{j \in J} z_j \leq n-2 \tag{30}$$

and obtain the following final formulations,

$$\min\{u | (6b) - (6e), (6h)(28), (29), (30)\},$$

which is presented in (21). In fact, without (30), MAE_a cannot be well-defined since it becomes negative for $p > n-1$ and the denominator becomes 0 for $p = n-1$. Observe that (21) is an MIP with $2n+4m+3$ variables (including m binary variables) and $n+5m+2$ constraints. Observe also that (6) with the additional constraint (30) can be used for the fat case. However, using $n-2$ explanatory variables out of m candidate explanatory variables can lead to an extremely small SAE as we explained at the beginning of this section.

To obtain a valid value of M for v_j's in (21), we can use a similar concept used in Section 2. In detail, we set

$$M := mae_0 + \frac{mae_0}{n-2} = \frac{n-1}{n-2} mae_0 \tag{31}$$

for v_j's to consider regression models that are better than having no regression variables. Given a heuristic solution with objective function value mae_a^{heur}, we can strengthen M by making solutions worse than the heuristic solution infeasible. Hence, we set $M := mae_a^{heur} + \frac{mae_0}{n-2}$ for v_j's in (29).

However, obtaining a valid value of M for x_j's in (21) is not trivial. Note that (12), which we used for the thin case, is not applicable for the fat case because LP (10) can easily be unbounded for the fat case. One valid procedure is to (i) generate all possible combinations of $n-2$ explanatory variables and all n observations, (ii) compute M for each combination using the procedure in Section 2.1.3, and (iii) pick the maximum value out of all possible combinations. However, this is a combinatorial problem. Actually, the computational complexity of this procedure is as much as that of solving (1) for all possible subsets. Hence, enumerating all possible subsets just to get a valid big M is not tractable.

Instead, we can use a heuristic approach to obtain a good estimation of the valid value of M. In Appendix B, we propose a statistic-based procedure that ensures a valid value of M with a certain confidence level. This procedure can give an M value that is valid with 95% confidence. However, for the instances considered in this paper, this procedure gives values of M that are too large because many columns can be strongly correlated to each other. Note that a large value of M can cause numerical errors when solving the MIP's.

Hence, for computational experiment, we use a simple heuristic approach instead. Let us assume that we are given a feasible solution to (21) from a heuristic and x_j^{heur}'s are the coefficient of the regression model. Then, we set

$$M := \max_{j \in J} |x_j^{heur}|. \tag{32}$$

Note that we cannot say that (32) is valid or valid with 95% confidence. If we use (21) with this M, we get a heuristic (even if (21) is solved optimally).

Similar to MAE_a in (20), MSE_a can be defined as

$$MSE_a = \frac{SSE + \frac{p}{n-2} mse_0}{n-1-p}, \tag{33}$$

where $mse_0 = \frac{\sum_{i \in I}(b_i - \bar{b})^2}{n-1}$ is the mean squared error of an optimal regression model when $p = 0$. Next, similar to (28) and (29), we define

$$v_j \leq u + \frac{mse_0}{n-2}, \tag{34}$$

$$u + \frac{mse_0}{n-2} - M(1-z_j) \leq v_j \leq M z_j, \tag{35}$$

while (7) remains the same. Finally, we obtain

$$\min\{u | (7), (6c) - (6e), (6h)(34), (35), (30)\} \tag{36}$$

for the MSE_a objective. Note that (36) is mixed integer quadratically constrained program that has $2n + 4m + 3$ variables and $n + 5m + 2$ constraints.

For the core set algorithm, similar to (23), we have

$$\min\{u | (7), (6c) - (6e), (6h), (34), (35)\}. \tag{37}$$