Journal of the Operations Research Society of Japan
Vol. 62, No. 1, January 2019, pp. 15–36

APPLICATION OF A MIXED INTEGER NONLINEAR PROGRAMMING APPROACH TO VARIABLE SELECTION IN LOGISTIC REGRESSION

Keiji Kimura
Kyushu University

(Received March 14, 2018; Revised September 25, 2018)

Abstract Variable selection is the process of finding variables relevant to a given dataset in model construction. One of the techniques for variable selection is exponentially evaluating many models with a goodness-of-fit (GOF) measure, for example, Akaike information criterion (AIC). The model with the lowest GOF value is considered as the best model. We proposed a mixed integer nonlinear programming approach to AIC minimization for linear regression and showed that the approach outperformed existing approaches in terms of computational time [13]. In this study, we apply the approach in [13] to AIC minimization for logistic regression and explain that a few of the techniques developed previously [13], for example, relaxation and a branching rule, can be used for the AIC minimization. The proposed approach requires solving relaxation problems, which are unconstrained convex problems. We apply an iterative method with an effective initial guess to solve these problems. We implement the proposed approach via SCIP, which is a noncommercial optimization software and a branch-and-bound framework. We compare the proposed approach with a piecewise linear approximation approach developed by Sato and others [16]. The results of computational experiments show that the proposed approach finds the model with the lowest AIC value if the number of candidates for variables is 45 or lower.

Keywords: Optimization, mixed integer nonlinear programming, SCIP optimization suite, Akaike information criterion, logistic regression, variable selection

1. Introduction

1.1. Variable selection in logistic regression

Finding the best statistical model for a given dataset is one of the most important problems in statistical applications (e.g., linear and logistic regression). This problem is called *variable selection,* and solving it leads to the following benefits: improvement in the prediction performance of a statistical model, development of faster and more cost-effective models in terms of computation, and better understanding of the essence of the statistical model behind a given dataset. See [11] for more details.

To evaluate statistical models comprised of selected variables, a few goodness-of-fit (GOF) measures, such as the Akaike information criterion (AIC) [2] and Bayesian information criterion (BIC) [17], are often employed. The goal of AIC-based variable selection is to find a model with the lowest AIC value among all the models. Because the number of all models is exponentially large, computation of all models is impractical. Instead of evaluating all models, stepwise methods are often applied. These methods are local search algorithms and procedures for finding statistical models with low AIC values. However, they may miss the model with the lowest AIC value.

Various approaches have been proposed for variable selection in logistic regression. ℓ_1-penalized logistic regression [14] is often employed because it provides sparse models and performs well even on large-scale instances. However, the models provided by this approach are not necessarily the best in terms of GOF measures. Sato and others formulated a mixed integer linear programming problem by employing a piecewise linear approximation to minimize GOF measures [16]. Although this approach might not arrive at the best statistical model, the results of their computational experiments indicated that this approach outperformed the stepwise methods. Bertsimas and King [6] proposed a mixed integer nonlinear programming (MINLP) approach to constructing models with the desired properties, for example, predictive power, interpretability, and sparsity. In addition, they proposed a tailored methodology using outer approximation techniques and dynamic constraint generation to solve the MINLP problem. The risk score problem [20] was optimized for feature selection, integer coefficient, and operational constraints. This problem was formulated as a MINLP problem that can be solved by using the cutting plane algorithm proposed in [20].

1.2. Mixed integer nonlinear programming

An MINLP can deal with integer variables and nonlinear functions and is one of the most flexible modeling paradigms from the viewpoint of formulation. However, this flexibility leads to numerical difficulties associated with the handling of nonlinear functions and challenges pertaining to optimization in the context of integrality. Nonetheless, many researchers and practitioners have shown interest in solving MINLP problems. Several methods have been proposed for solving MINLP problems, for example, branch-and-bound (B&B) algorithm, branch-and-cut algorithm, outer approximation, and Benders decomposition. See [5, 21] for details about MINLP.

The availability and maturity of software for solving MINLP problems have increased significantly in the past 20 years. A number of open sources and commercial MINLP solvers are listed in [9]. A customized MINLP solver for a specific application occasionally achieves good computational performance [8, 10]. Herein, we solve an MINLP problem for variable selection and implement a few techniques for the problem by customizing the SCIP Optimization Suite [18]. This software toolbox comprises several parts, such as SCIP [1, 21] and UG [19]. SCIP is open source software, and it provides a B&B framework for solving mixed integer linear programming and MINLP problems. Additional plugins, such as branching rules, relaxation handlers, and primal heuristics, allow for an efficient solution process. UG provides a parallel extension of SCIP to employ multi-threaded parallel computation. These software applications have been developed by the Optimization Department at the Zuse Institute Berlin and its collaborators.

1.3. Contributions and structure of the present paper

In the present study, we apply an MINLP approach to AIC-based variable selection in logistic regression. In [13], we proposed the MINLP approach for minimizing AIC in the context of linear regression. The MINLP approach executes a B&B algorithm and requires that a relaxation problem is solved at each B&B node. When the MINLP approach is applied to AIC minimization for logistic regression, the relaxation problem becomes an unconstrained convex problem that can be solved by applying an iterative method, for example, the steepest descent method and Newton's method. To reduce the computational time for solving the relaxation problem, we develop an effective procedure to construct an initial guess for the iterative method.

We proposed a few techniques pertaining to the B&B algorithm in [13], for example, relaxation, a branching rule, and heuristics based on stepwise methods. These techniques can be applied to AIC minimization for logistic regression, and they perform well in terms of computational time. We implement these techniques by customizing SCIP [1, 21], which is a mathematical optimization software and a B&B framework. The results of computational experiments show that the proposed approach finds the model with the lowest AIC value if the number of candidates for variables is 45 or lower.

In addition, we explain that the proposed MINLP approach can be used for ℓ_0-penalized variable selection, that is

$$\min_{\beta \in \mathbb{R}^p} \quad f(\beta) + \lambda \|\beta\|_0. \tag{1.1}$$

Here, λ is a positive constant, and $\|\beta\|_0$ is the ℓ_0-norm of β, that is, the count of the nonzero elements in β. The function f represents a discrepancy between a given dataset and a statistical model. The proposed MINLP approach can be applied to the problem (1.1) if the proposed relaxation problem can be solved at each B&B node.

The remainder of this paper is organized as follows. In Section 2, we briefly introduce AIC minimization for logistic regression and formulate it as an MINLP problem. In Section 3, we show that the techniques proposed in [13] can be applied to the formulated problem. In Section 4.1, we develop the procedure for constructing the initial guess of the iterative method. In Section 4.2, we briefly explain a piecewise linear approximation approach [16] for logistic regression to compare the proposed approach with the said approach. In Section 4.3, we report numerical experiments conducted using the proposed approach, piecewise linear approximation approach, and stepwise methods. In Section 4.4, we examine which of the proposed techniques is effective and how our heuristics method and branching rule influence changes in upper and lower bounds of the optimal value. In Section 5, we explain how the proposed MINLP approach can be applied to ℓ_0-penalized variable selection.

2. AIC Minimization for Logistic Regression

In this section, we formulate AIC minimization for logistic regression as a MINLP problem. First, we define a logistic regression model and AIC. Logistic regression is a fundamental statistical tool, and it estimates the probability of a binary response from a given dataset $(x_{i1}, \ldots, x_{ip}, y_i) \in \mathbb{R}^p \times \{0, 1\}$ with $x_{i1} = 1$ $(i = 1, \ldots, n)$. We regard y_i as a class label of the ith data for all $i = 1, \ldots, n$. Logistic regression determines coefficient parameters $\beta_1, \ldots, \beta_p$ of the following logistic regression model which determines the probability of $y = 1$ for an input $x = (x_1, \ldots, x_p)^T \in \mathbb{R}^p$,

$$P(y = 1 \mid x) = \frac{\exp\left(\sum_{j=1}^p \beta_j x_j\right)}{1 + \exp\left(\sum_{j=1}^p \beta_j x_j\right)}.$$

Here $x_1, \ldots, x_p$ and y are explanatory variables and a response variable, respectively. The probability of $y = 0$ is obtained by simple calculation,

$$P(y = 0 \mid x) = 1 - P(y = 1 \mid x) = \frac{1}{1 + \exp\left(\sum_{j=1}^p \beta_j x_j\right)}.$$

Therefore, the probability of $y \in \{0, 1\}$ can be written as

$$P(y \mid x) = \frac{\exp\left(y \sum_{j=1}^p \beta_j x_j\right)}{1 + \exp\left(\sum_{j=1}^p \beta_j x_j\right)}.$$

In logistic regression, the coefficient parameters $\beta_1, \ldots, \beta_p$ can be determined by maximum likelihood estimation. In fact, the log-likelihood function ℓ is defined as

$$\ell(\beta) = \sum_{i=1}^{n} \log P(y_i \mid x^i) = -\sum_{i=1}^{n} \left(\log\left(1 + \exp\left(\beta^T x^i\right)\right) - y_i \beta^T x^i\right),$$

where $\beta = (\beta_1, \ldots, \beta_p)^T$ and $x^i = (x_{i1}, \ldots, x_{ip})^T$ for $i = 1, \ldots, n$.

The AIC [2] is one of GOF measures, and it can evaluate logistic regression models. Let $\{1, \ldots, p\}$ be a set of indices of given explanatory variables and S a subset of $\{1, \ldots, p\}$. For any subset $S \subseteq \{1, \ldots, p\}$, the AIC value of the logistic regression model with the jth explanatory variables ($j \in S$) can be computed as follows:

$$\begin{aligned} \mathrm{AIC}(S) = 2 \min_{\beta_j} & \left\{ \sum_{i=1}^{n} \left(\log\left(1 + \exp\left(\beta^T x^i\right)\right) - y_i \beta^T x^i\right) : \begin{array}{l} \beta_j = 0 \ (j \in \{1, \ldots, p\} \setminus S) \\ \beta \in \mathbb{R}^p \end{array} \right\} \\ & + 2\|\beta^*\|_0, \end{aligned} \tag{2.1}$$

where β^* is an optimal solution of the minimization problem in (2.1), and $\|\beta^*\|_0$ is the ℓ_0-norm of β^*, that is, the count of the nonzero elements in β^*. The objective function of the minimization in (2.1) is convex because its Hessian matrix is positive semidefinite. The minimization in (2.1) is solved for any subset $S \subseteq \{1, \ldots, p\}$ by applying a gradient algorithm, for instance, the steepest descent method and Newton's method.

In AIC-based variable selection, the logistic regression model with the lowest AIC value is selected as the best model. It is practically difficult to compute the AIC value (2.1) for all models because the number of models is 2^p. Hence, we apply an efficient MINLP approach to finding the best model. The minimization of AIC(S) over $S \subseteq \{1, \ldots, p\}$ is formulated as the following MINLP problem:

$$\min_{\beta, z} \ 2\sum_{i=1}^{n} \left(\log\left(1 + \exp\left(\beta^T x^i\right)\right) - y_i \beta^T x^i\right) + 2\sum_{j=1}^{p} z_j \tag{2.2}$$

$$\text{s.t.} \ z_j = 0 \Rightarrow \beta_j = 0 \ (j = 1, \ldots, p), \tag{2.3}$$

$$\beta_j \in \mathbb{R}, \ z_j \in \{0, 1\} \ (j = 1, \ldots, p). \tag{2.4}$$

The constraints (2.3) represent indicator constraints, that is, β_j has to be zero if z_j is zero.

3. Solving the MINLP Problem (2.2)–(2.4)

In [13], we formulated AIC minimization for linear regression as an MINLP problem and proposed a B&B algorithm purpose-built for this problem. The algorithm consists of components related to effective relaxation, handling of data structure, a heuristic method, and a branching rule. In this section, we explain how these components are applicable to AIC minimization for logistic regression. In Section 5, we describe how they are applicable to ℓ_0-penalized variable selection (1.1).

The first term of the objective function (2.2) is denoted by $f(\beta)$, that is,

$$f(\beta) := 2\sum_{i=1}^{n} \left(\log\left(1 + \exp\left(\beta^T x^i\right)\right) - y_i \beta^T x^i\right). \tag{3.1}$$

We develop a method based on a B&B algorithm [1, 21] to solve the problem (2.2)–(2.4). The B&B algorithm splits repeatedly a set of *feasible solutions* into two sets by *branching* and constructs a *B&B tree* of the *node* corresponding to the split set. At each node, the algorithm computes a *lower bound* of the optimal value of a *subproblem* by *relaxation.* In Section 3.1, we describe the relaxation to compute the lower bounds efficiently. Moreover, we show that a feasible solution of the subproblem can be obtained easily from an optimal solution of the proposed relaxation problem. In Sections 3.2 and 3.3, we describe a few techniques to improve the numerical performance.

3.1. Relaxation to compute lower bounds

We explain how relaxation proposed in [13] can be applied to the problem (2.2)–(2.4). Branching fixes a binary variable z_j of the problem (2.2)–(2.4) to zero or one and generates two nodes repeatedly. For any node, we define the sets Z_0, Z_1, and Z as follows:

$$\begin{aligned} Z_1 &= \{j \in \{1, \dots, p\} : z_j \text{ is already fixed to } 1\}, \\ Z_0 &= \{j \in \{1, \dots, p\} : z_j \text{ is already fixed to } 0\}, \\ Z &= \{j \in \{1, \dots, p\} : z_j \text{ is not fixed}\}. \end{aligned}$$

Then, the subproblem of the problem (2.2)–(2.4) can be expressed as follows:

$$\begin{aligned} \min_{\beta, z} \quad & f(\beta) + 2\sum_{j=1}^{p} z_j && (3.2) \\ \text{s.t.} \quad & \beta_j \in \mathbb{R},\ z_j = 1\ (j \in Z_1),\ \beta_j = z_j = 0\ (j \in Z_0), && (3.3) \\ & z_j = 0 \Rightarrow \beta_j = 0,\ \beta_j \in \mathbb{R},\ z_j \in \{0, 1\}\ (j \in Z). && (3.4) \end{aligned}$$

We denote the subproblem (3.2)–(3.4) by $Q(Z_1, Z_0, Z)$ because the subproblem can be specified uniquely by using Z_1, Z_0, and Z. By relaxing the integrality of the variables z_j, we obtain the following standard relaxation problem of $Q(Z_1, Z_0, Z)$:

$$\begin{aligned} \min_{\beta, z} \quad & f(\beta) + 2\sum_{j=1}^{p} z_j && (3.5) \\ \text{s.t.} \quad & \beta_j \in \mathbb{R},\ z_j = 1\ (j \in Z_1),\ \beta_j = z_j = 0\ (j \in Z_0), && (3.6) \\ & z_j = 0 \Rightarrow \beta_j = 0,\ \beta_j \in \mathbb{R},\ 0 \le z_j \le 1\ (j \in Z). && (3.7) \end{aligned}$$

The optimal value of the problem (3.5)–(3.7) is the lower bound of the optimal value of $Q(Z_1, Z_0, Z)$. Instead of solving (3.5)–(3.7), we consider the following problem:

$$\min_{\beta}\ f(\beta) + 2\#(Z_1) \text{ s.t. } \beta_j = 0\ (j \in Z_0),\ \beta_j \in \mathbb{R}\ (Z_1 \cup Z), \tag{3.8}$$

where $\#(Z_1)$ stands for the number of elements in the set Z_1. This problem is arrived at by eliminating the indicator constraints and the variables z_j from the problem (3.5)–(3.7). Notably, the optimal value of the problem (3.8) is the lower bound of the optimal value of $Q(Z_1, Z_0, Z)$. In fact, the optimal value of (3.8) is smaller than or equal to the optimal value of (3.5)–(3.7) because any feasible solution (β, z) of (3.5)–(3.7) is also feasible for (3.8) and satisfies the following inequality:

$$f(\beta) + 2\sum_{j=1}^{p} z_j = f(\beta) + 2\left(\sum_{j \in Z} z_j + \#(Z_1)\right) \ge f(\beta) + 2\#(Z_1).$$

Hence, we employ (3.8) as a relaxation problem of the subproblem $Q(Z_1, Z_0, Z)$ to compute a lower bound of the optimal value of $Q(Z_1, Z_0, Z)$. We denote the relaxation problem (3.8) by $R(Z_1, Z_0, Z)$, which is an unconstrained convex problem. We can solve $R(Z_1, Z_0, Z)$ under the practical assumption of logistic regression analysis. See Section 5 and Appendix A for details. In the numerical experiments conducted herein, we obtain an optimal solution of $R(Z_1, Z_0, Z)$ by applying Newton's method.

We show the following lemma that implies the optimal value of $R(Z_1, Z_0, Z)$ is identical to the optimal value of the standard relaxation problem (3.5)–(3.7).

Lemma 3.1. *Let θ^* be the optimal value of $R(Z_1, Z_0, Z)$. Then, the optimal value of (3.5)–(3.7) is θ^*.*

Proof. Let β^* be an optimal solution of $R(Z_1, Z_0, Z)$. We construct a sequence $\{(\beta^N, z^N)\}_N^\infty$ as follows:

$$\beta^N = \beta^* \text{ and } z_j^N = \begin{cases} 1 & \text{if } j \in Z_1, \\ 1/N & \text{if } j \in Z, \quad (j = 1, \ldots, p) \\ 0 & \text{otherwise,} \end{cases}$$

for all $N \geq 1$. (β^N, z^N) is feasible for (3.5)–(3.7) for all $N \geq 1$. It is sufficient to prove that the objective value θ^N of (3.5)–(3.7) at (β^N, z^N) converges to the optimal value θ^* of $R(Z_1, Z_0, Z)$ as N approaches infinity. Because we have $\theta^* = f(\beta^*) + 2\#(Z_1)$ and

$$\theta^* \leq \theta^N = f(\beta^*) + 2\#(Z_1) + \frac{2}{N}\#(Z) = \theta^* + \frac{2}{N}\#(Z),$$

θ^N converges to θ^* as N approaches to infinity. This implies that the optimal value of $R(Z_1, Z_0, Z)$ is identical to the optimal value of (3.5)–(3.7). □

We can easily solve the relaxation problem of the subproblem obtained by fixing z_j to 1. By fixing the variable z_k, two subproblems $Q(Z_1 \cup \{k\}, Z_0, Z \backslash \{k\})$ and $Q(Z_1, Z_0 \cup \{k\}, Z \backslash \{k\})$ are generated from $Q(Z_1, Z_0, Z)$. The relaxation problem $R(Z_1 \cup \{k\}, Z_0, Z \backslash \{k\})$ can then be formulated as follows:

$$\min_{\beta} \; f(\beta) + 2\#(Z_1 \cup \{k\}) \text{ s.t. } \beta_j = 0 \; (j \in Z_0), \;\; \beta_j \in \mathbb{R} \; (Z_1 \cup Z).$$

Therefore, the optimal value of the relaxation problem $R(Z_1 \cup \{k\}, Z_0, Z \backslash \{k\})$ for any $k \in Z$ is $\theta^* + 2$, where θ^* is the optimal value of the relaxation problem $R(Z_1, Z_0, Z)$.

We explain a procedure to generate a feasible solution of the subproblem $Q(Z_1, Z_0, Z)$ from an optimal solution of $R(Z_1, Z_0, Z)$. Let $\hat{\beta} = (\hat{\beta}_1, \ldots, \hat{\beta}_p)^T$ be the optimal solution of $R(Z_1, Z_0, Z)$. We define $\hat{z} = (\hat{z}_1, \ldots, \hat{z}_p)$ by $\hat{z}_j = 1$ if $\hat{\beta}_j \neq 0$, otherwise $\hat{z}_j = 0$. Clearly, $(\hat{\beta}, \hat{z})$ is feasible for $Q(Z_1, Z_0, Z)$.

3.2. Effective handling of data structure

Standard statistical textbooks often assume that datasets have linear independence; however, as it is some datasets in the UCI Machine Learning Repository [4], for example, `bumps` and `stat-G`, have linear dependence. Given that we apply Newton's method to the relaxation problem (3.8), it is necessary to solve linear systems. If a given dataset has linear dependence, the linear systems may have infinitely many solutions. In other words, the function $f(\beta)$ is not strongly convex. Hence, we explain the processing of linear dependence in logistic regression and use the idea of the processing proposed in [13].

First, we explain the following proposition, which involves techniques for solving (2.2)–(2.4).

Proposition 3.2. *Let S be a nonempty subset of $\{1,\dots,p\}$. We assume that for any $s \in S$ and $\tilde{\beta} = (\tilde{\beta}_1,\dots,\tilde{\beta}_p)^T \in \mathbb{R}^p$, there exists $\hat{\beta} \in \mathbb{R}^p$ such that*

$$\hat{\beta}_j = \tilde{\beta}_j \ (j \in \{1,\dots,p\}\backslash S),\ \hat{\beta}_s = 0 \text{ and } f(\tilde{\beta}) = f(\hat{\beta}),$$

where the function f is defined in (3.1). Then, the following properties are satisfied:

1. *If $S \subseteq Z_1$, the subproblem $Q(Z_1, Z_0, Z)$ is pruned in the B&B tree, that is, the optimal value of $Q(Z_1, Z_0, Z)$ is larger than the optimal value of (2.2)–(2.4).*
2. *If $Z \cap S \neq \emptyset$ and $S \subseteq Z_1 \cup Z$, the optimal value of the relaxation problem $R(Z_1, Z_0, Z)$ is equal to the optimal value of the relaxation problem $R(Z_1, Z_0 \cup \{k\}, Z\backslash\{k\})$ for any $k \in Z \cap S$.*

We prove Proposition 3.2 at the end of this subsection.

Remark. The first property of Proposition 3.2 implies that we can reduce the number of generated B&B nodes. The second property of Proposition 3.2 implies that we can reduce the computational cost of solving the relaxation problem. In fact, we can remove a continuous variable β_k $(k \in Z \cup S)$ from the relaxation problem, where the set S satisfies the assumption in Proposition 3.2. We apply this removal repeatedly. Therefore, we can efficiently solve (2.2)–(2.4) by using the properties of Proposition 3.2.

Next, we show that f defined in (3.1) satisfies the assumption in Proposition 3.2 if a given dataset has linear dependence. To explain this, we define linear dependence in datasets. For a given dataset $(x_{i1},\dots,x_{ip},y_i) \in \mathbb{R}^p \times \{0,1\}$ with $x_{i1} = 1$ $(i = 1,\dots,n)$, we define the following vectors:

$$x_j = \begin{pmatrix} x_{1j} \\ \vdots \\ x_{nj} \end{pmatrix} \in \mathbb{R}^n \text{ for } j = 1,\dots,p.$$

If these vectors $x_1,\dots,x_p \in \mathbb{R}^n$ are linearly dependent, we say that the dataset has linearly dependent variables. Lemmas 3.3 and 3.4 show that linear dependence in a given dataset corresponds to the assumption in Proposition 3.2. Hence, we can reduce the computational cost by applying Proposition 3.2.

Lemma 3.3. *If a given dataset has linearly dependent variables, there exists a nonempty set $S \subseteq \{1,\dots,p\}$ such that*

$$\sum_{j \in S} \alpha_j x_j = 0 \text{ and } \alpha_j \neq 0 \text{ for all } j \in S. \tag{3.9}$$

Proof. If a given dataset has linearly dependent variables, there exists α $(\neq 0) \in \mathbb{R}^p$ such that $\sum_{j=1}^p \alpha_j x_j = 0$. Then, the subset S is defined by $\{j \in \{1,\dots,p\} : \alpha_j \neq 0\}$. It is readily apparent that S is nonempty. □

Lemma 3.4. *If a given dataset has linearly dependent variables, there exists a nonempty set $S \subseteq \{1,\dots,p\}$ such that the S and f defined in (3.1) satisfy the assumption in Proposition 3.2.*

Proof. Let $\tilde{\beta}$ be $(\tilde{\beta}_1,\dots,\tilde{\beta}_p)^T \in \mathbb{R}^p$ and I_p a set $\{1,\dots,p\}$. From Lemma 3.3, there exists a nonempty set $S \subseteq \{1,\dots,p\}$ such that (3.9). We consider the two cases: (i) $\#(S) = 1$ and (ii) $\#(S) > 1$.

(i). If S contains a single element (i.e., $S = \{s\}$), $x_s = 0$. We define $\hat{\beta} = (\hat{\beta}_1, \ldots, \hat{\beta}_p)^T \in \mathbb{R}^p$ as follows:

$$\hat{\beta}_j = \begin{cases} \tilde{\beta}_j, & (j \in I_p \backslash \{s\}) \\ 0, & (j = s) \end{cases}$$

for all $j = 1, \ldots, p$. Because $\tilde{\beta}^T x^i = \hat{\beta}^T x^i$, $f(\tilde{\beta}) = f(\hat{\beta})$ is satisfied.
(ii). For any $s \in S$, there exist $\alpha'_j \neq 0$ $(j \in S \backslash \{s\})$ such that

$$x_{is} = \sum_{j \in S \backslash \{s\}} \alpha'_j x_{ij}$$

for all $i = 1, \ldots, n$. $\tilde{\beta}^T x^i$ $(i = 1, 2, \ldots, n)$ can be written as follows:

$$\begin{aligned} \tilde{\beta}^T x^i &= \sum_{j \in I_p \backslash \{s\}} \tilde{\beta}_j x_{ij} + \tilde{\beta}_s x_{is} \\ &= \sum_{j \in I_p \backslash \{s\}} \tilde{\beta}_j x_{ij} + \tilde{\beta}_s \sum_{j \in S \backslash \{s\}} \alpha'_j x_{ij} \\ &= \sum_{j \in I_p \backslash S} \tilde{\beta}_j x_{ij} + \sum_{j \in S \backslash \{s\}} (\tilde{\beta}_j + \tilde{\beta}_s \alpha'_j) x_{ij}. \end{aligned}$$

Here, we define $\hat{\beta} = (\hat{\beta}_1, \ldots, \hat{\beta}_p)^T \in \mathbb{R}^p$ as follows:

$$\hat{\beta}_j = \begin{cases} \tilde{\beta}_j, & (j \in I_p \backslash S) \\ \tilde{\beta}_j + \tilde{\beta}_s \alpha'_j, & (j \in S \backslash \{s\}) \\ 0, & (j = s) \end{cases}$$

for all $j = 1, \ldots, p$. Because $\tilde{\beta}^T x^i = \hat{\beta}^T x^i$, $f(\tilde{\beta}) = f(\hat{\beta})$ is satisfied. □

As described at the start of this subsection, the linear system appearing in Newton's method may have infinitely many solutions if a relaxation problem $R(Z_1, Z_0, Z)$ has sets of linearly dependent vectors. Therefore, we transform $R(Z_1, Z_0, Z)$ to eliminate such sets. To this end, we use the second property of Proposition 3.2. We describe the nonempty set $S \subseteq \{1, \ldots, p\}$ of Lemma 3.3 as a linearly dependent set. Given any relaxation problem $R(Z_1, Z_0, Z)$ and a linearly dependent set $S \subseteq Z_1 \cup Z$ with $Z \cap S \neq \emptyset$, we select an index $k \in Z \cap S$ and solve $R(Z_1, Z_0 \cup \{k\}, Z \setminus \{k\})$ instead of $R(Z_1, Z_0, Z)$. Because $R(Z_1, Z_0 \cup \{k\}, Z \setminus \{k\})$ does not contain the vector $x_k \in \mathbb{R}^n$, it is regarded as a problem without the linearly dependent set S. Hence, application of the second property of Proposition 3.2 corresponds to removal of the linearly dependent set from $R(Z_1, Z_0, Z)$.

To apply Proposition 3.2 at each B&B node, we must find linearly dependent sets. In Algorithm 1, we describe a process proposed in [13] to find a collection $\mathcal{C}(Z, Z_1)$ of the linearly dependent sets. This process ensures that Proposition 3.2 is available for any nonempty set $S \in \mathcal{C}(Z, Z_1)$. We state that the linear system (3.10) has a unique solution because the matrix $(x_k)_{k \in S}$ has full column rank. To save computational costs, we find $\mathcal{C}(\{1, \ldots, p\}, \emptyset)$ in advance and reuse it. If the intersection of all linearly dependent sets of a given dataset is $\emptyset$, then it is sufficient to find $\mathcal{C}(\{1, \ldots, p\}, \emptyset)$. In fact, it contains all linearly dependent sets in the given dataset. Otherwise, the linear system may yield infinitely many solutions with Newton's method even after application of the second property of Proposition 3.2 with

Algorithm 1: An algorithm to find a collection of linearly dependent sets

```
Input: vectors $x_j$ $(j \in Z \cup Z_1)$
Output: A collection $\mathcal{C}(Z, Z_1)$ of linearly dependent sets
$\mathcal{C}(Z, Z_1) \longleftarrow \emptyset,\ S \longleftarrow \emptyset$;
for $j \in Z \cup Z_1$ do
    if the vectors $\{x_k : k \in S \cup \{j\}\}$ are linearly independent then
        $S \longleftarrow S \cup \{j\}$;
    else
        Solve the following linear system:
            $\sum_{k \in S} \alpha_k x_k = x_j$        (3.10)
        $S' \longleftarrow \{k \in S : \alpha_k \neq 0\} \cup \{j\},\ \mathcal{C}(Z, Z_1) \longleftarrow \mathcal{C}(Z, Z_1) \cup \{S'\}$;
    end
end
return $\mathcal{C}(Z, Z_1)$
```

$\mathcal{C}(\{1, \dots, p\}, \emptyset)$ to $R(Z_1, Z_0, Z)$. In this case, we alternate between executing Algorithm 1 and applying the second property.

Finally, we prove Proposition 3.2 as follows:

Proof. (*First property of Proposition 3.2*). Let m_Q be the optimal value of $Q(Z_1, Z_0, Z)$ and m_P the optimal value of the problem (2.2)–(2.4). It is sufficient to prove that $m_Q > m_P$. An optimal solution of $Q(Z_1, Z_0, Z)$ is denoted by $(\tilde{\beta}, \tilde{z}) \in \mathbb{R}^p \times \mathbb{R}^p$. Considering the assumption of this proposition, for $s \in S$, there exists $\hat{\beta} \in \mathbb{R}^p$ such that

$$\hat{\beta}_j = \tilde{\beta}_j \ (j \in \{1, \dots, p\} \backslash S),\ \hat{\beta}_s = 0 \textit{ and } f(\tilde{\beta}) = f(\hat{\beta}).$$

We define $\hat{z} = (\hat{z}_1, \dots, \hat{z}_p)^T \in \{0, 1\}^p$ as follows:

$$\hat{z}_j = \begin{cases} \tilde{z}_j, & (\text{if } j \neq s) \\ 0, & (\text{if } j = s) \end{cases}$$

for all $j = 1, \dots, p$. Because $\tilde{z}_s$ is one and $(\hat{\beta}, \hat{z})$ is feasible for (2.2)–(2.4),

$$m_Q = f(\tilde{\beta}) + 2\sum_{j=1}^{p} \tilde{z}_j > f(\hat{\beta}) + 2\sum_{j=1}^{p} \hat{z}_j \geq m_P.$$

(*Second property of Proposition 3.2*). Let m_R be the optimal value of the relaxation problem $R(Z_1, Z_0, Z)$ and m_{R_k} the optimal value of the relaxation problem $R(Z_1, Z_0 \cup \{k\}, Z \backslash \{k\})$ for $k \in Z \cap S$. m_R and m_{R_k} are computed as follows:

$$\begin{aligned} m_R &= \min_{\beta} \{f(\beta) + 2\#(Z_1) : \beta \in \mathbb{R}^p, \beta_j = 0 \ (j \in Z_0)\}, \\ m_{R_k} &= \min_{\beta} \{f(\beta) + 2\#(Z_1) : \beta \in \mathbb{R}^p, \beta_j = 0 \ (j \in Z_0 \cup \{k\})\}. \end{aligned}$$

Because an optimal solution of the relaxation problem $R(Z_1, Z_0 \cup \{k\}, Z \backslash \{k\})$ is feasible for the relaxation problem $R(Z_1, Z_0, Z)$, $m_{R_k} \geq m_R$ is satisfied. Let $\tilde{\beta}$ be an optimal solution of the relaxation problem $R(Z_1, Z_0, Z)$. Considering the assumption of this proposition, there exists $\hat{\beta} \in \mathbb{R}^p$ such that

$$\hat{\beta}_j = \tilde{\beta}_j \ (j \in \{1, \ldots, p\} \backslash S), \ \hat{\beta}_k = 0 \textit{ and } f(\tilde{\beta}) = f(\hat{\beta}).$$

Because $\hat{\beta}$ is feasible for the relaxation problem $R(Z_1, Z_0, \cup\{k\}, Z\backslash\{k\})$, $m_R \geq m_{R_k}$ is satisfied. Hence $m_R = m_{R_k}$. □

3.3. The other techniques to improve computational performance

By customizing SCIP [1, 21], we realize the relaxation problem (3.8) and Proposition 3.2. In addition, we employ a heuristic method and a branching rule, which are described in our paper [13]. In this section, we briefly introduce these techniques and explain how they can be applied to AIC minimization in logistic regression (i.e., (2.2)–(2.4)).

To prune B&B nodes from a B&B tree, it is necessary to find a good feasible solution early. SCIP contains many heuristic methods for finding feasible solutions of MINLP problems [21]. However, these methods do not always find feasible solutions of (2.2)–(2.4). Our heuristic method is based on stepwise methods with forward selection and backward elimination. In each step, the stepwise methods decide whether to add an explanatory variable to the statistical model or to remove it. This process is repeated until no further improvement is possible. These stepwise methods are implemented in statistical software, for example, R [15]. Although these methods are considered local search algorithms, they often find good statistical models within a short time. We extend the capability of the stepwise methods to find feasible solutions of any subproblem $Q(Z_1, Z_0, Z)$. As a result, we expect that our heuristic methods will find good feasible solutions early. We describe the heuristic method for (2.2)–(2.4) in Algorithm 2. For any subset $S \subseteq \{1, \ldots, p\}$ with $Z_1 \subseteq S \subseteq Z_1 \cup Z$, we define the value $\bar{\theta}^S$ and the vector $\bar{z}^S = (\bar{z}_1^S, \ldots, \bar{z}_p^S)^T \in \{0,1\}^p$ as follows:

$$\bar{\theta}^S := \min_{\beta} \{f(\beta) : \beta_j = 0 \ (j \in \{1, \ldots, p\} \setminus S), \ \beta \in \mathbb{R}^p\} + 2\#(S), \tag{3.11}$$

$$\bar{z}_j^S := \begin{cases} 1 & \text{if } j \in S \\ 0 & \text{if } j \in \{1, \ldots, p\} \setminus S \end{cases} \quad \text{for all } j = 1, \ldots, p, \tag{3.12}$$

where $\#(S)$ denotes the number of elements in S. The vector $\bar{\beta}^S \in \mathbb{R}^p$ denotes an optimal solution of the minimization problem in (3.11). We use $(\bar{\beta}^{Z_1}, \bar{z}^{Z_1})$ and $(\bar{\beta}^{Z_1 \cup Z}, \bar{z}^{Z_1 \cup Z})$ as the initial solutions (β^1, z^1) and (β^2, z^2), respectively, in our implementation. In Section 4, we show that our heuristic method improves computational performance.

A branching rule selects a branching variable at each node. Because branching is one of the cores of the B&B algorithm, it is important for solving MINLP problems to find good strategies. See [1, Section 5] for details about branching rules. We employ *most frequent branching*, which was proposed in [13]. This branching rule is based on two tendencies: some explanatory variables are often employed in good statistical models and are adopted in the best statistical model. By branching variables z_k, which correspond to such explanatory variables, good feasible solutions might be eliminated from the generated subproblem (3.2)–(3.4) with $z_k = 0$. Hence, we expect that the subproblem is pruned as early as possible. We describe the branching rule in Algorithm 3. In Section 4, we compare this rule numerically with *inference branching* implemented in SCIP and observe that most frequent branching is more effective than inference branching for the benchmark datasets.

Algorithm 2: Our heuristics based on the stepwise methods

```
Input: A subproblem $Q(Z_1, Z_0, Z)$ and two initial feasible solutions $(\beta^1, z^1)$ and
       $(\beta^2, z^2)$ of $Q(Z_1, Z_0, Z)$
Output: A feasible solution $(\beta, z)$ of $Q(Z_1, Z_0, Z)$
$S \longleftarrow \{j \in \{1, \dots, p\} : z^1_j = 1\}$, $v^f \longleftarrow \infty$;
/* the stepwise method with forward selection                                   */
while $\bar{\theta}^S < v^f$ do
    $v^f \longleftarrow \bar{\theta}^S$, $(\beta^f, z^f) \longleftarrow (\bar{\beta}^S, \bar{z}^S)$;
    Find $J = \arg\min_{j \in Z \setminus S}\{\bar{\theta}^{S \cup \{j\}} : \bar{z}^{S \cup \{j\}}$ is feasible for $Q(Z_1, Z_0, Z)\}$;
    if $J = \emptyset$ then break;
    Select $j \in J$ and $S \longleftarrow S \cup \{j\}$;
end
$S \longleftarrow \{j \in \{1, \dots, p\} : z^2_j = 1\}$, $v^b \longleftarrow \infty$;
/* the stepwise method with backward elimination                                */
while $\bar{\theta}^S < v^b$ do
    $v^b \longleftarrow \bar{\theta}^S$, $(\beta^b, z^b) \longleftarrow (\bar{\beta}^S, \bar{z}^S)$;
    Find $J = \arg\min_{j \in Z \cap S}\{\bar{\theta}^{S \setminus \{j\}} : \bar{z}^{S \setminus \{j\}}$ is feasible for $Q(Z_1, Z_0, Z)\}$;
    if $J = \emptyset$ then break;
    Select $j \in J$ and $S \longleftarrow S \setminus \{j\}$;
end
if $v^f < v^b$ then return $(\beta^f, z^f)$;
else return $(\beta^b, z^b)$;
```

Algorithm 3: Most frequent branching

```
Input: A positive integer $N$, a set $Z$ of indices of unfixed variables, and the current
       pool of feasible solutions of (2.2)–(2.4)
Output: A branching variable $z_k$ $(k \in Z)$
Choose the top $N$ feasible solutions $(\beta^1, z^1), \dots, (\beta^N, z^N)$ from the pool;
/* Here $(\beta^i, z^i)$ is a feasible solution with the $i$th lowest objective
   value in the pool.                                                           */
for $j \in Z$ do
    Compute score value $s_j := \sum_{i=1}^{N} z^i_j$;
end
return $z_k$ with $s_k = \max_{j \in Z}\{s_j\}$
```

4. Numerical Experiments

4.1. A developed solver for the problem (2.2)–(2.4)

We discussed the techniques that can be used in conjunction with the B&B algorithm to efficiently solve the problem (2.2)–(2.4) in Section 3. We implement these techniques by customizing SCIP [1, 21], which provides a framework of the B&B algorithm. Moreover, we execute multi-threaded parallel computation via UG [19], which provides a parallel extension of SCIP.

At each B&B node, the solver developed herein computes the optimal value of the proposed relaxation problem $R(Z_1, Z_0, Z)$

$$\min_{\beta}\ 2\sum_{i=1}^{n}\left(\log\left(1+\exp\left(\beta^T x^i\right)\right) - y_i\beta^T x^i\right) + 2\#(Z_1) \text{ s.t. } \beta_j = 0\ (j \in Z_0),\ \beta_j \in \mathbb{R}\ (Z_1 \cup Z),$$

by applying Newton's method. This method is iterative, and it requires an initial feasible solution of the relaxation problem. The developed solver constructs the initial feasible solution from an optimal solution of the relaxation problem of the parent node. To explain this procedure, we focus on two relaxation problems $R(Z_1 \cup \{k\}, Z_0, Z\backslash\{k\})$ and $R(Z_1, Z_0 \cup \{k\}, Z\backslash\{k\})$, which are obtained by fixing the variable z_k. Then, the relaxation problem of the parent node is $R(Z_1, Z_0, Z)$. Let θ^* be the optimal value of $R(Z_1, Z_0, Z)$ and $\beta^* = (\beta_1^*, \dots, \beta_p^*)^T \in \mathbb{R}^p$ the optimal solution of $R(Z_1, Z_0, Z)$. In Section 3.1, we showed that the optimal value of $R(Z_1 \cup \{k\}, Z_0, Z\backslash\{k\})$ is $\theta^* + 2$. The initial feasible solution $\beta^0 = (\beta_1^0, \dots, \beta_p^0)^T \in \mathbb{R}^p$ of the other relaxation problem $R(Z_1, Z_0 \cup \{k\}, Z\backslash\{k\})$ can be constructed as follows:

$$\beta_j^0 = \begin{cases} 0 & \text{if } j = k, \\ \beta_j^* & \text{otherwise,} \end{cases}$$

for all $j = 1, \dots, p$. Because β^* is feasible for $R(Z_1, Z_0, Z)$, β^0 is feasible for $R(Z_1, Z_0 \cup \{k\}, Z\backslash\{k\})$. In Section 4.4, we show that this procedure reduces computational time.

4.2. A piecewise linear approximation approach [16]

Sato and others proposed an approach to variable selection for logistic regression analysis [16]. Their approach employs a piecewise linear approximation and a mixed integer linear programming problem. The greatest advantage of their approach is that commercial optimization software (e.g., CPLEX [12]) can be used to solve the mixed integer linear programming problem. Their approach can be applied to AIC minimization for logistic regression (i.e., the problem (2.2)–(2.4)). In Section 4.3, we compare the developed solver with their piecewise linear approximation approach.

We briefly explain the piecewise linear approximation approach to solving the problem (2.2)–(2.4). For a given dataset $(x_{i1}, \dots, x_{ip}, y_i) \in \mathbb{R}^p \times \{0, 1\}$ with $x_{i1} = 1$ $(i = 1, \dots, n)$, we define sets I_1 and I_2 as follows:

$$I_1 = \{i \in \{1, \dots, n\} : y_i = 1\} \text{ and } I_0 = \{i \in \{1, \dots, n\} : y_i = 0\}.$$

The function $F(\beta, z)$ denotes the objective function (2.2), and it can be rewritten as follows:

$$\begin{aligned} F(\beta, z) &:= 2\sum_{i=1}^{n}\left(\log\left(1+\exp\left(\beta^T x^i\right)\right) - y_i\beta^T x^i\right) + 2\sum_{j=1}^{p} z_j \\ &= 2\sum_{i\in I_1}\left(\log\left(1+\exp\left(\beta^T x^i\right)\right) - \beta^T x^i\right) + 2\sum_{i\in I_0}\log\left(1+\exp\left(\beta^T x^i\right)\right) + 2\sum_{j=1}^{p} z_j \\ &= 2\sum_{i\in I_1}\log\left(1+\exp\left(-\beta^T x^i\right)\right) + 2\sum_{i\in I_0}\log\left(1+\exp\left(\beta^T x^i\right)\right) + 2\sum_{j=1}^{p} z_j. \end{aligned}$$

We define the function $g(v)$ as $g(v) := \log\left(1 + \exp\left(-v\right)\right)$ and rewrite it as

$$F(\beta, z) = 2\sum_{i\in I_1} g(\beta^T x^i) + 2\sum_{i\in I_0} g(-\beta^T x^i) + 2\sum_{j=1}^{p} z_j.$$

By introducing extra variables $t_i (i = 1, \dots, n)$, the problem (2.2)–(2.4) can be reformulated as follows:

$$\min_{\beta,z} \quad 2\sum_{i=1}^{n} t_i + 2\sum_{j=1}^{p} z_j \tag{4.1}$$

$$\text{s.t.} \quad t_i \ge g(\beta^T x^i) \ (i \in I_1), \ t_i \ge g(-\beta^T x^i) \ (i \in I_0), \tag{4.2}$$

$$z_j = 0 \Rightarrow \beta_j = 0, \ \beta_j \in \mathbb{R}, \ z_j \in \{0,1\} \ (j = 1, \dots, p). \tag{4.3}$$

Given any set of points $V = \{v_1, \dots, v_K\}$, we can construct a relaxation problem of (4.1)–(4.3) by using the convexity of g

$$\min_{\beta,z} \quad 2\sum_{i=1}^{n} t_i + 2\sum_{j=1}^{p} z_j \tag{4.4}$$

$$\text{s.t.} \quad t_i \ge g'(v_k)(\beta^T x^i - v_k) + g(v_k) \ (i \in I_1; \ v_k \in V), \tag{4.5}$$

$$t_i \ge -g'(v_k)(\beta^T x^i + v_k) + g(v_k) \ (i \in I_0; \ v_k \in V), \tag{4.6}$$

$$z_j = 0 \Rightarrow \beta_j = 0, \ \beta_j \in \mathbb{R}, \ z_j \in \{0,1\} \ (j = 1, \dots, p). \tag{4.7}$$

The problem (4.4)–(4.7) is a mixed integer linear programming problem, and it can be solved by using standard optimization software. The optimal value $\bar{\theta}$ of (4.4)–(4.7) is a lower bound of the optimal value θ^* of (2.2)–(2.4). Let $(\bar{\beta}, \bar{z}, \bar{t})$ be an optimal solution of (4.4)–(4.7). We can construct the logistic regression model from the set of the selected explanatory variables $\bar{S} = \{j \in \{1, \dots, p\} : \bar{z}_j = 1\}$. Then, the AIC value of the constructed model is $\mathrm{AIC}(\bar{S})$. Hence, we obtain the following inequality:

$$\bar{\theta} \le \theta^* \le \mathrm{AIC}(\bar{S}).$$

If $\mathrm{AIC}(\bar{S}) - \bar{\theta}$ is small, the constructed model is guaranteed to be of good quality.

In the numerical experiments, we employ the following two sets as V,

$$V_1 = \{0, \pm 0.89, \pm 1.90, \pm 3.55, \pm\infty\},$$
$$V_2 = \{0, \pm 0.44, \pm 0.89, \pm 1.37, \pm 1.90, \pm 2.63, \pm 3.55, \pm 5.16, \pm\infty\}.$$

These sets can be computed by using the greedy algorithm proposed in [16].

4.3. Comparison with the piecewise linear approximation approach and stepwise methods

In this subsection, we show numerical experiments* pertaining to AIC minimization for logistic regression and compare the developed solver with the piecewise linear approximation approach and the stepwise methods. We use benchmark datasets from the UCI Machine Learning Repository [4] and standardize the datasets to have zero mean and unit variance.

Table 1 shows a comparison of the performance of the following methods:

- MINLP:
  - refers to the proposed approach implemented in SCIP [1, 21] and UG [19],
  - executes the B&B algorithm by using the techniques described in Sections 3 and 4.1,
  - uses 16 threads for parallel computation.

*The specifications of the computer used in the numerical experiments are as follows: CPU: Intel® Xeon® CPU E5–2687 @ 3.1GHz; Memory: 128GB; and OS: Ubuntu 16.04.3 LTS

- SW_+:
  - refers to the stepwise method starting with no explanatory variables,
  - is implemented by C++ and LAPACK [3].
- SW_-:
  - refers to the stepwise method starting with all explanatory variables,
  - is implemented by C++ and LAPACK [3].
- MILP(V):
  - refers to the piecewise linear approximation approach [16] with the point set V,
  - solves the mixed integer linear programming problem (4.4)–(4.7) with CPLEX [12],
  - employs the better of the two solutions of the stepwise methods as the initial solution.
  - employs 16 threads for parallel computation.

The columns labeled "n," "p," and "k" indicate the number of data points, candidates for explanatory variables, and selected explanatory variables, respectively. The column labeled "AIC" indicates the computed AIC value. The AIC values in bold font are the best among the five values. The column labeled "obj_{MILP}" presents the objective value of the computed solution of the mixed integer linear programming problem (4.4)–(4.7). The column labeled "Time(sec)" indicates CPU time in seconds to compute the optimal value. ">5000" implies that the corresponding method could not determine the optimal value within 5000 seconds. The column labeled "Gap(%)" indicates the optimality gap used in SCIP, and it is defined as

$$\text{Gap} = \frac{|\text{upper bound} - \text{lower bound}|}{\min\{|\text{upper bound}|, |\text{lower bound}|\}} \times 100.$$

It can be inferred from Table 1 that MINLP outperforms MILP(V) in terms of computational time. In fact, for $p \leq 45$, MINLP was faster than both the MILP(V). Moreover, MINLP found the lowest AIC values of the five approaches on large-scale instances. However, for $p \geq 62$, even MINLP could not guarantee optimum within 5000 seconds.

4.4. Computational performance of the developed techniques

To examine which of the proposed techniques is effective, we present the computational performance of the following methods:

- MINLP:
  - executes the most frequent branching described in Section 3.3,
  - executes the heuristic method described in Section 3.3,
  - constructs the initial feasible solution from an optimal solution of the relaxation problem of the parent node.
  - executes the procedure developed in Section 4.1 to construct the initial guess for Newton's method.
- $\text{MINLP}_{\text{w/o-mfb}}$:
  - corresponds to MINLP without the most frequent branching,
  - executes the inference branching in SCIP.
- $\text{MINLP}_{\text{w/o-heur}}$: corresponds to MINLP without the heuristic method.
- $\text{MINLP}_{\text{w/o-guess}}$:
  - corresponds to MINLP without the initial guess,
  - employs the zero vector as a initial feasible solution.

Table 1: Comparison of the proposed method with the piecewise linear approximation approach and the stepwise methods

Name	n	p	Methods	AIC	obj$_{\text{MILP}}$	k	Time(sec)	Gap(%)
bumps	2584	22	MINLP	**1097.11**	—	9	20.08	0.00
			SW$_+$	1097.37	—	9	0.92	—
			SW$_-$	1100.66	—	13	0.54	—
			MILP(V_1)	1098.12	1060.51	8	41.51	0.00
			MILP(V_2)	1099.98	1086.43	9	627.36	0.00
breast-P	194	34	MINLP	**147.04**	—	19	25.76	0.00
			SW$_+$	162.94	—	13	0.24	—
			SW$_-$	152.13	—	25	0.25	—
			MILP(V_1)	**147.04**	144.56	19	112.40	0.00
			MILP(V_2)	**147.04**	146.40	19	279.15	0.00
biodeg	1055	42	MINLP	**653.29**	—	23	221.54	0.00
			SW$_+$	654.79	—	25	2.01	—
			SW$_-$	**653.29**	—	23	2.25	—
			MILP(V_1)	**653.29**	640.75	23	>5000	0.93
			MILP(V_2)	**653.29**	649.62	23	>5000	2.39
spectf	267	45	MINLP	**168.33**	—	15	432.45	0.00
			SW$_+$	172.34	—	10	0.36	—
			SW$_-$	169.42	—	17	0.79	—
			MILP(V_1)	169.34	163.54	14	515.74	0.00
			MILP(V_2)	169.34	165.53	14	1603.12	0.00
stat-G	1000	62	MINLP	**958.15**	—	24	>5000	5.54
			SW$_+$	**958.15**	—	24	3.09	—
			SW$_-$	963.70	—	29	2.55	—
			MILP(V_1)	**958.15**	944.50	24	>5000	5.21
			MILP(V_2)	**958.15**	954.46	24	>5000	5.10
musk	6598	166	MINLP	**1706.89**	—	115	>5000	16.55
			SW$_+$	1733.56	—	120	292.18	—
			SW$_-$	**1706.89**	—	115	609.44	—
			MILP(V_1)	**1706.89**	1663.02	115	>5000	16.68
			MILP(V_2)	**1706.89**	1693.28	115	>5000	16.39
madelon	2000	500	MINLP	**2502.06**	—	105	>5000	20.76
			SW$_+$	2504.02	—	102	316.92	—
			SW$_-$	2905.58	—	422	>5000	—
			MILP(V_1)	2504.02	2471.93	102	>5000	20.20
			MILP(V_2)	2504.02	2493.70	102	>5000	22.85

The column labeled "Nodes" in Table 2 indicates the number of generated B&B nodes. To indicate the effective techniques, we underline the highest values among all the methods in Table 2. We observe the following from Table 2:

- For $p \leq 45$, MINLP, that is, the developed solver incorporating all techniques, was the fastest among the four methods. This implies that the most frequent branching, the heuristic method based on the stepwise methods, and the initial guess are effective for solving (2.2)–(2.4).
- MINLP and $\text{MINLP}_{\text{w/o-guess}}$ could solve AIC minimization for `spectf` within 5000 seconds. However, $\text{MINLP}_{\text{w/o-mfb}}$ and $\text{MINLP}_{\text{w/o-heur}}$ could not solve the minimization within 5000 seconds. Hence, the most frequent branching and the heuristic method based on the stepwise methods are more effective than the initial guess in this instance.
- For $p \geq 62$, $\text{MINLP}_{\text{w/o-heur}}$ were the worst among the four methods in terms of solution quality. Hence, it is evident from this result that the heuristic method described in Section 3.3 is an important technique for large-scale instances.

We examine how the heuristic method and the most frequent branching described in Section 3.3 influence changes in the upper and lower bounds. Figure 1 shows the results of the upper bounds for `biodeg` and `spectf`. The solid and the broken lines correspond to our solver with and without the heuristic method based on the stepwise methods (i.e., MINLP and $\text{MINLP}_{\text{w/o-heur}}$), respectively. Our solver with the heuristic method immediately found good feasible solutions compared to the solver without the heuristic method. Figure 2 shows the results of the lower bounds for `biodeg` and `spectf`. The solid and the broken lines correspond to our solver with and without the most frequent branching (i.e., MINLP and $\text{MINLP}_{\text{w/o-mfb}}$), respectively. Our solver without the most frequent branching appears to stop increases in the lower bounds halfway. The benefit of using the most frequent branching can be confirmed from Figure 2.

5. An Extension of Our MINLP Approach

In variable selection based on optimization, an objective function typically consists of two competing terms (see, e.g., [11]): *the goodness-of-fit* and *the number of explanatory variables.* In this section, we consider the following MINLP formulation for variable selection:

$$\min_{\beta,z} \quad f(\beta) + \lambda \sum_{j=1}^{p} z_j \tag{5.1}$$

$$\text{s.t.} \quad z_j = 0 \Rightarrow \beta_j = 0 \ (j = 1, \dots, p), \tag{5.2}$$

$$\beta_j \in \mathbb{R}, \ z_j \in \{0,1\} \ (j = 1, \dots, p), \tag{5.3}$$

where $\beta = (\beta_1, \dots, \beta_p)^T$ represents the parameters in a given statistical model, and λ is a positive constant. The first term $f(\beta)$ of the objective function (5.1) corresponds to the goodness-of-fit, for example, a discrepancy between the given dataset and the statistical model. The second term $\lambda \sum_{j=1}^{p} z_j$ operates as a penalty for the number of variables. This problem (5.1)–(5.3) is considered ℓ_0-penalized variable selection. We assume the following for $f(\beta)$ in the objective function (5.1):

Assumption 1. *For any nonempty subset $S \subseteq \{1, \dots, p\}$, we can compute the optimal value and an optimal solution of the following optimization problem:*

$$\min_{\beta \in \mathbb{R}^p} \ f(\beta) \ \textit{s.t.} \ \beta_j = 0 \ (j \in \{1, \dots, p\} \backslash S). \tag{5.4}$$

Table 2: The computational performance of our developed techniques

Name	n	p	Methods	AIC	k	Time(sec)	Nodes	Gap(%)
bumps	2584	22	MINLP	1097.11	9	20.08	3.6×10^3	0.00
			MINLP$_{\text{w/o-mfb}}$	1097.11	9	44.99	2.2×10^4	0.00
			MINLP$_{\text{w/o-heur}}$	1097.11	9	28.68	$\underline{2.3 \times 10^4}$	0.00
			MINLP$_{\text{w/o-guess}}$	1097.11	9	$\underline{46.48}$	4.1×10^3	0.00
breast-P	194	34	MINLP	147.04	19	25.76	1.5×10^5	0.00
			MINLP$_{\text{w/o-mfb}}$	147.04	19	$\underline{554.07}$	$\underline{3.3 \times 10^6}$	0.00
			MINLP$_{\text{w/o-heur}}$	147.04	19	31.87	4.6×10^5	0.00
			MINLP$_{\text{w/o-guess}}$	147.04	19	27.38	1.5×10^5	0.00
biodeg	1055	42	MINLP	653.29	23	221.54	1.7×10^5	0.00
			MINLP$_{\text{w/o-mfb}}$	653.29	23	$\underline{>5000}$	$\underline{8.8 \times 10^6}$	$\underline{4.53}$
			MINLP$_{\text{w/o-heur}}$	653.29	23	1018.83	2.5×10^6	0.00
			MINLP$_{\text{w/o-guess}}$	653.29	23	586.45	1.9×10^5	0.00
spectf	267	45	MINLP	168.33	15	432.45	1.1×10^6	0.00
			MINLP$_{\text{w/o-mfb}}$	168.33	15	$\underline{>5000}$	$\underline{1.1 \times 10^7}$	29.89
			MINLP$_{\text{w/o-heur}}$	$\underline{171.80}$	17	$\underline{>5000}$	$\underline{1.1 \times 10^7}$	$\underline{34.53}$
			MINLP$_{\text{w/o-guess}}$	168.33	15	574.13	1.5×10^5	0.00
stat-G	1000	62	MINLP	958.15	24	>5000	7.7×10^6	5.54
			MINLP$_{\text{w/o-mfb}}$	958.15	24	>5000	6.5×10^6	6.11
			MINLP$_{\text{w/o-heur}}$	$\underline{978.67}$	30	>5000	5.5×10^6	$\underline{7.61}$
			MINLP$_{\text{w/o-guess}}$	958.15	24	>5000	$\underline{8.9 \times 10^6}$	4.62
musk	6598	166	MINLP	1706.89	115	>5000	3.5×10^4	16.55
			MINLP$_{\text{w/o-mfb}}$	1705.01	111	>5000	5.7×10^4	16.87
			MINLP$_{\text{w/o-heur}}$	$\underline{1774.54}$	161	>5000	$\underline{6.4 \times 10^5}$	$\underline{20.18}$
			MINLP$_{\text{w/o-guess}}$	1706.89	115	>5000	2.1×10^4	17.19
madelon	2000	500	MINLP	2502.06	105	>5000	1.0×10^6	20.76
			MINLP$_{\text{w/o-mfb}}$	2503.58	105	>5000	1.1×10^6	21.15
			MINLP$_{\text{w/o-heur}}$	$\underline{3028.85}$	455	>5000	$\underline{2.4 \times 10^6}$	$\underline{46.70}$
			MINLP$_{\text{w/o-guess}}$	2502.06	105	>5000	8.3×10^5	20.76

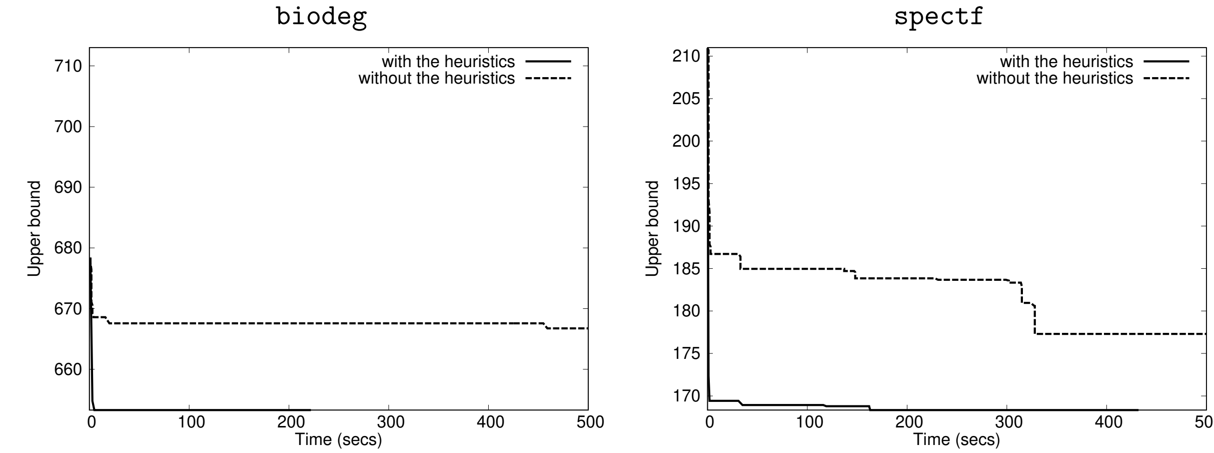

Figure 1: The evolution of the upper bounds in the first 500 seconds, for `biodeg` and `spectf` when using our solver with and without our heuristic method

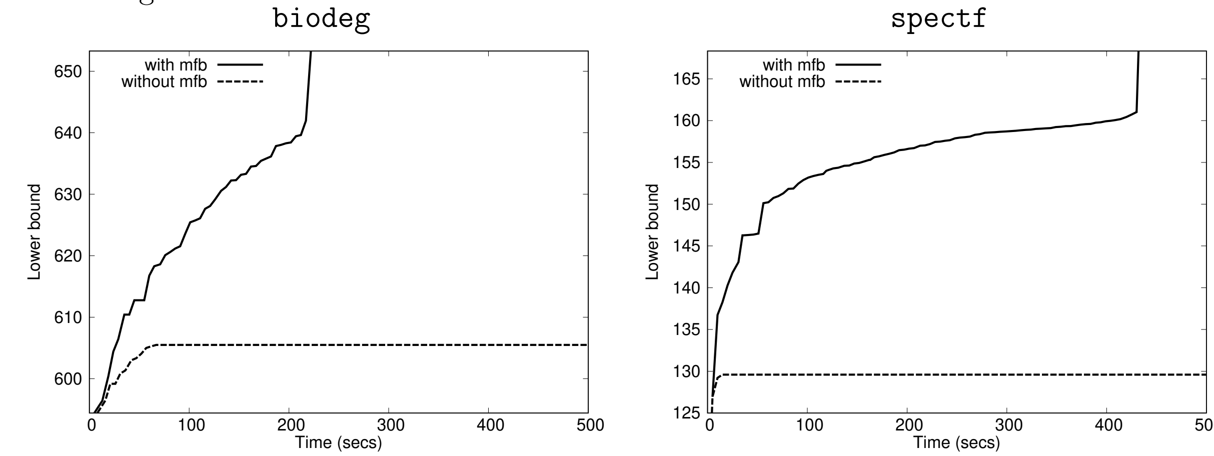

Figure 2: The evolution of the lower bounds in the first 500 seconds, for `biodeg` and `spectf` when using our solver with and without the most frequent branching

If f is a strongly convex function, the problem (5.4) becomes an unconstrained convex problem that can be solved by applying a gradient algorithm, for instance, the steepest descent method and Newton's method. The AIC minimization for logistic regression (i.e., the problem (2.2)–(2.4)) is of the form of the problem (5.1)–(5.3). Assumption 1 holds for the logistic regression analysis under the practical assumption. In other words, Assumption 1 fails in a certain dataset. See Appendix A for more details.

In Section 3, we defined $f(\beta)$ as the first term of the objective function (2.2) and discussed the following techniques for the numerical performance:

(i) the relaxation problem (3.8),

(ii) the two properties of Proposition 3.2,

(iii) the heuristic method described in Algorithm 2,

(iv) the most frequent branching described in Algorithm 3.

These techniques can be applied to the problem (5.1)–(5.3) if the first term $f(\beta)$ of (5.1) satisfies Assumption 1. The reasons for this are as follows: (i) and (iii) Assumption 1 implies that we can compute the optimal values of the proposed relaxation problem (3.8) and the optimization problem (3.11) for the heuristic method; (ii) the two properties can be applied if the function f and a nonempty subset $S \subseteq \{1, \dots, p\}$ satisfy the assumption in

Proposition 3.2; and (iv) the most frequent branching does not depend on the form of the function f.

6. Conclusion

We applied the MINLP approach to AIC-based variable selection in logistic regression and showed that the techniques proposed in [13] can be applied to the variable selection. In addition to these techniques, the developed solver can construct an effective initial guess to increase computational performance in terms of solving the relaxation problem. In the numerical experiments, the most frequent branching, the heuristic method based on the stepwise methods, and the initial guess were effective in terms of computational time. If the number of candidates of explanatory variables was 45 or lower, our solver could find the models with the lowest AIC values. Moreover, our solver outperformed the piecewise linear approximation approach employing high standard optimization software.

We developed a solver for the problem (2.2)–(2.4) by using SCIP and UG, which provide a flexible framework of a B&B algorithm and parallel computation [18]. For small-scale and medium-scale instances, our solver showed good computational performance because of the customization of SCIP and UG for the specific problem. Conversely, for large-scale instances, there is room for improvement in the numerical performance of our solver. The computational cost of our heuristic method based on the stepwise methods appears to be high for large instances. In fact, SW_+ and SW_- (i.e., the stepwise methods) required considerably more computational time for solving `musk` and `madelon` compared to the small-scale and medium-scale instances. Hence, further study is to reduce the computational time of our heuristic method, for example, by applying discrete first order algorithms [7].

In Section 5, we explained that the proposed MINLP approach can be applied to ℓ_0-penalized variable selection. By changing the objective function in (5.1), other information criteria, for example, the Bayesian information criterion and the Hannan-Quinn information criterion, can be employed to evaluate logistic regression models. Furthermore, the problem (5.1)–(5.3) can handle linear regression and basis function regression as well. Considering these findings, it can be inferred that our solver is flexible in terms of formulation.

Acknowledgements

I would like to thank Hayato Waki for useful discussions and carefully reading the manuscript. I am grateful to the referees for their helpful comments.

References

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

[2] H. Akaike: A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, **19** (1974), 716–723.

[3] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen: *LAPACK Users' Guide, Third Edition* (Society for Industrial and Applied Mathematics, 1991). LAPACK home page: `http://www.netlib.org/lapack`

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

[5] P. Belotti, C. Kirches, S. Leyffer, J. Linderoth, J. Luedtke, and A. Mahajan: Mixed-integer nonlinear optimization. *Acta Numerica*, **22** (2013), 1–131.

[6] D. Bertsimas and A. King: Logistic regression: from art to science. *Statistical Science*, **32** (2017), 367–384.

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

[8] C. Bragalli, C. D'Ambrosio, J. Lee, A. Lodi, and P. Toth: On the optimal design of water distribution networks: a practical MINLP approach. *Optimization and Engineering*, **13** (2012), 219–246.

[9] M.R. Bussiek and S. Vigerske: MINLP solver software. In J.J. Cochran, L.A., Cox, P. Keskinocak, J.P., Kharoufeh, and J.C., Smith (eds.): *Wiley Encyclopedia of Operations Research and Management Science* (Wiley Online Library, 2014).

[10] T. Farkas, B. Czuczai, E. Rev, and Z. Lelkes: New MINLP model and modified outer approximation algorithm for distillation column synthesis. *Industrial and Engineering Chemistry Research*, **47** (2008), 3088–3103.

[11] I. Guyon and A. Elisseeff: An introduction to variable and feature selection. *Journal of Machine Learning Research*, **3** (2003), 1157–1182.

[12] IBM ILOG CPLEX Optimizer 12.8.0. (IBM ILOG, 2017).
CPLEX home page: `https://www.ibm.com/products/ilog-cplex-optimization-studio`

[13] K. Kimura and H. Waki: Minimization of Akaike's information criterion in linear regression analysis via mixed integer nonlinear program. *Optimization Methods and Software*, **33** (2018), 633–649.

[14] K. Koh, S. Kim, and S. Boyd: An interior-point method for large-scale ℓ_1-regularized logistic regression. *Journal of Machine Learning Research*, **8** (2007), 1519–1555.

[15] R. Ihaka and R. Gentleman: R: a language and environment for statistical computing. *Journal of Computational and Graphical Statistics*, **5** (1996), 299–314.
R home page: `http://www.R-project.org`

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

[17] G. Schwarz: Estimating the dimension of a model. *Annals of Statistics*, **6** (1978), 461–464.

[18] SCIP Optimization Suite 5.0.0. (Zuse Institute Berlin, 2017).
SCIP home page: `http://scip.zib.de/`

[19] Ubiquity Generator framework. (Zuse Institute Berlin, 2017).
UG home page: `http://ug.zib.de/`

[20] B. Ustun and C. Rudin: Learning optimized risk scores. arXiv preprint, arXiv:1610.00168 (2018).

[21] S. Vigerske and A. Gleixner: SCIP: Global optimization of mixed-integer nonlinear programs in a branch-and-cut framework. *Optimization Methods and Software*, **33** (2018), 563–593.

A. When Does Logistic Regression Satisfy Assumption 1?

As mentioned in Section 5, the MINLP formulation (2.2)–(2.4) for logistic regression may not satisfy Assumption 1. Therefore, here, we provide a necessary and sufficient condition to ensure that the MINLP formulation (2.2)–(2.4) for logistic regression satisfies Assumption 1. First, we introduce notation and symbols. For a dataset $(x^i, y_i) \in \mathbb{R}^p \times \{0,1\}$ $(i = 1, \dots, n)$, we define the sets I_1 and I_0 as

$$I_1 = \{i \in \{1,\dots,n\} : y_i = 1\} \text{ and } I_0 = \{i \in \{1,\dots,n\} : y_i = 0\}.$$

We rewrite the objective function of the minimization (5.4) as

$$f(\beta) = \sum_{i \in I_0} \log\left(1 + \exp(\beta^T x^i)\right) + \sum_{i \in I_1} \log\left(1 + \exp(-\beta^T x^i)\right).$$

For $\beta \in \mathbb{R}^p$, we define the sets $J_+(\beta), J_-(\beta)$, and $J_0(\beta)$ as

$$\begin{aligned}&J_+(\beta) = \{i \in \{1,\dots,n\} : \beta^T x^i > 0\},\ J_-(\beta) = \{i \in \{1,\dots,n\} : \beta^T x^i < 0\} \text{ and}\\ &J_0(\beta) = \{i \in \{1,\dots,n\} : \beta^T x^i = 0\}.\end{aligned}$$

Then, we have $J_\bullet(\gamma\beta) = J_\bullet(\beta)$ for $\gamma > 0$ and $\bullet \in \{+,-,0\}$. For any $\gamma > 0$ and $\beta \in \mathbb{R}^p$, we have

$$\begin{aligned}f(\gamma\beta) &= \sum_{i \in I_0} \log\left(1 + \exp(\gamma\beta^T x^i)\right) + \sum_{i \in I_1} \log\left(1 + \exp(-\gamma\beta^T x^i)\right)\\ &= \sum_{i \in I_0 \cap J_+(\beta)} \log\left(1 + \exp(\gamma\beta^T x^i)\right) + \sum_{i \in I_0 \cap J_-(\beta)} \log\left(1 + \exp(\gamma\beta^T x^i)\right)\\ &+ \sum_{i \in I_1 \cap J_+(\beta)} \log\left(1 + \exp(-\gamma\beta^T x^i)\right) + \sum_{i \in I_1 \cap J_-(\beta)} \log\left(1 + \exp(-\gamma\beta^T x^i)\right)\\ &+ \#(J_0(\beta)) \log(2). \end{aligned} \tag{A.1}$$

It follows from the following theorem that Assumption 1 holds when the necessary and sufficient condition in the theorem holds.

Theorem A.1. *The minimization (5.4) has an optimal solutions for any nonempty subset $S \subseteq \{1,\dots,p\}$ if and only if for any $\beta \in \mathbb{R}^p \setminus \{0\}$, $I_0 \cap J_+(\beta)$ or $I_1 \cap J_-(\beta)$ is nonempty.*

Proof. For simplicity, we fix $S = \{1,\dots,p\}$ for (5.4). First, we prove the if part. We fix $\beta \in \mathbb{R}^p$ so that $\|\beta\| = 1$. Then, by taking $\gamma \to \infty$, each term in (A.1) satisfies

$$\begin{aligned}&\sum_{i \in I_0 \cap J_+(\beta)} \log\left(1 + \exp(\gamma\beta^T x^i)\right) \to +\infty, \sum_{i \in I_0 \cap J_-(\beta)} \log\left(1 + \exp(\gamma\beta^T x^i)\right) \to 0,\\ &\sum_{i \in I_1 \cap J_+(\beta)} \log\left(1 + \exp(-\gamma\beta^T x^i)\right) \to 0, \sum_{i \in I_1 \cap J_-(\beta)} \log\left(1 + \exp(-\gamma\beta^T x^i)\right) \to +\infty.\end{aligned}$$

Because we have assumed that $I_0 \cap J_+(\beta)$ or $I_1 \cap J_-(\beta)$ is nonempty, there exists $M > 0$ such that the objective function $f(\beta)$ takes sufficiently large values for all β so that $\|\beta\| > M$. Hence, the minimum solution of (5.4) is in the circle $\|\beta\| \le M$. Therefore, (5.4) has an optimal solution.

Next, we prove the only-if part. We assume that there exists $\beta \in \mathbb{R}^p \setminus \{0\}$ such that both $I_0 \cap J_+(\beta)$ and $I_1 \cap J_-(\beta)$ are empty. It is sufficient to prove that (5.4) has a finite optimal value but no optimal solutions. It follows from the definition of $f(\beta)$ that $f(\beta) > \#(J_0(\beta)) \log(2)$ for all $\beta \in \mathbb{R}^p \setminus \{0\}$. In addition, from the proof of the if-part, by taking $\gamma \to \infty$, we have $g(\gamma\beta) \to \#(J_0(\beta)) \log(2)$. This is the desired result. □

Keiji Kimura
Graduate School of Mathematics
Kyushu University
744 Motooka, Nishi-ku,
Fukuoka, 819-0395, Japan
E-mail: k-kimura@math.kyushu-u.ac.jp