数理解析研究所講究録
第2027巻 2017年 60-69

変数選択のための混合整数非線形計画法

九州大学大学院数理学府　木村 圭児
九州大学マス・フォア・インダストリ研究所　脇 隼人
Keiji Kimura
Faculty of Mathematics, Kyushu University
Hayato Waki
Institute of Mathematics, Kyushu University

1　はじめに

統計学におけるモデル推定では, 情報量規準を用いてモデルに必要なパラメータの取捨選択を行うことがある. これは変数選択と呼ばれ, 情報量規準として, 赤池情報量規準 (Akaike Information Criterion: AIC) などが用いられる. 推定されたモデルに対し AIC は計算され, AIC が小さい方が良いモデルである. AIC が最小となるようなパラメータを決定することで, 観測データとの当てはまりの良さを損なわないように予測精度の良いモデルを推定することができる. AIC ができるだけ小さい値をとる変数選択の一般的な手法の一つとして, ステップワイズ法が知られている. ステップワイズ法は, R[11] などの既存の統計ソフトウェアに実装されていて, 計算が高速で広く用いられている. しかし, ステップワイズ法は局所探索をしているとみなせるので, 選択された変数の組合せに対する AIC が最小であるとは限らない. したがって, 本研究で変数選択のための混合整数非線形計画法を提案する.

混合整数非線形計画法は, 非線形関数と整数変数を扱うことができるため, 柔軟な定式化が可能な最適化手法である. 混合整数非線形計画法を用いた豊富な応用先がある一方で, 最適化ソルバーの性能は発展途上であり, 近年盛んに研究されている.

線形回帰における変数選択に対して, 混合整数二次錐計画問題を用いる手法が提案されている [8]. この手法は最適性が保証され, 変数の数が 30 個以下であれば, 現実的な時間で求解できる. この他に, 線形回帰における変数選択の手法として, 混合整数二次計画問題を用いる手法も提案されている [3].

本論文では, [7] の概略を述べる. [7] では, 線形回帰における AIC 最小化に対し, [8] に現れる問題を基に混合整数非線形計画問題として定式化し, 定式化された問題を効率良く解くための手法を提案している. 分枝限定法のフレームワークを提供している SCIP (Solving Constraint Integer Program [1, 10, 14]) を用いて, 提案する手法を実装する. SCIP は自由度が高いソフトウェアであり, 解法を細かく制御するプログラムを実装することができる. 5 節で, 線形回帰における AIC 最小化に対する, 提案する手法と既存手法を比較する.

2 AIC最小化

2.1 AIC

赤池情報量規準 (Akaike Information Criterion: AIC)[2] は, 推定されたモデルを評価するための基準の一つであり, 次の式で与えられる.

$$\begin{aligned}\mathrm{AIC} &= -2(\text{モデルの最大対数尤度}) + 2(\text{モデルの自由パラメータ数}) \\ &= -2\max_{\boldsymbol{\beta}}\{\ell(\boldsymbol{\beta}) : \boldsymbol{\beta} \in \mathbb{R}^k\} + 2k. \end{aligned} \tag{1}$$

ただし, $\ell(\boldsymbol{\beta})$ は対数尤度関数, k はモデルの自由パラメータ数を表す.

AIC を評価基準とする変数選択では, 与えらた説明変数の候補からいくつかの説明変数を用いてモデルを推定し, 推定されたモデルの AIC を計算する. 説明変数の候補の集合を $\{1,\ldots,p\}$ とし, 説明変数の候補 $j \in \{1,\ldots,p\}$ に対応するパラメータを β_j とする. このとき, 任意の部分集合 $S \subseteq \{1,\ldots,p\}$ に対する AIC は, (1) を用いて次のように計算できる.

$$\mathrm{AIC}(S) = -2\max_{\boldsymbol{\beta}}\{\ell(\boldsymbol{\beta}) : \beta_j = 0\ (j \in \{1,\ldots,p\}\backslash S),\ \beta \in \mathbb{R}^{p+k'}\} + 2(\#(S) + k'). \tag{2}$$

ただし, k' はパラメータ β_j $(j = 1,\ldots,p)$ を除く自由パラメータの数であり, $\#(S)$ は S に含まれる要素の数を表す. AIC の値が小さい方が良いモデルであり, AIC の値が最小であるモデルは最適なモデルとして選択される. この変数選択の手法は AIC 最小化と呼ばれ, 次の問題として表すことができる.

$$\min_{S}\{\mathrm{AIC}(S) : S \subseteq \{1,\ldots,p\}\}.$$

説明変数の候補の組合せが 2^p 個あるため, 全てのモデルの AIC の値を計算し, AIC 最小化を行うことは現実的では無い. 2.2 節では, 線形回帰における AIC 最小化を効率良く行うために, 混合整数非線形計画問題と呼ばれる最適化問題に定式化する.

2.2 線形回帰における AIC 最小化

線形回帰モデルは, 現象の結果と複数の要因を関係づける基本的な統計モデルであり, 与えれたデータから係数パラメータ β_j を決定し, 次の式で表される.

$$y = \beta_0 + \sum_{j=1}^{p} \beta_j x_j.$$

$x_1,\ldots,x_p$ は説明変数と呼ばれる.

与えれた n 個のデータ

$$(y_i; x_{i1},\ldots,x_{ip}),\ i = 1,\ldots,n$$

から変数選択を行うとき, 説明変数の候補の添字集合は $\{1,\ldots,p\}$ であり, j 番目 $(j \in \{1,\ldots,p\})$ の説明変数の候補を選択しない場合は $\beta_j = 0$ である. $\boldsymbol{\beta} = (\beta_0, \beta_1,\ldots,\beta_p)^T \in \mathbb{R}^{p+1}$ とし, モデルとデータの誤差を $\epsilon_i = y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij}$ $(i = 1,\ldots,n)$ とおく. 各

ϵ_i $(i=1,\ldots,n)$ は互いに独立で, 平均 0, 分散 σ^2 の正規分布に従うとする. このとき, 対数尤度関数 $\ell(\boldsymbol{\beta},\sigma^2)$ は次の式で与えられる.

$$\ell(\boldsymbol{\beta},\sigma^2) = -\frac{n}{2}\log 2\pi - \frac{n}{2}\log\sigma^2 - \frac{1}{2\sigma^2}\sum_{i=1}^{n}\epsilon_i^2.$$

対数尤度関数 $\ell(\boldsymbol{\beta},\sigma^2)$ において, $\boldsymbol{\beta}$ を任意に固定して, $\lambda := \sigma^2$ の関数として微分すると,

$$\frac{d\ell}{d\lambda} = -\frac{n}{2\lambda} + \frac{1}{2\lambda^2}\sum_{i=1}^{n}\epsilon_i^2,\ \frac{d^2\ell}{d\lambda^2} = \frac{n}{2\lambda^2} - \frac{1}{\lambda^3}\sum_{i=1}^{n}\epsilon_i^2,$$

となるので, $\sigma^2 = (\sum_{i=1}^{n}\epsilon_i^2)/n$ のとき, $\ell(\boldsymbol{\beta},\sigma^2)$ 極大値かつ最大値をとる. したがって, (2) より, 任意の部分集合 $S \subseteq \{1,\ldots,p\}$ に対する AIC の値は次のように計算できる.

$$\begin{aligned}\mathrm{AIC}(S) = \min_{\boldsymbol{\beta}} &\left\{ n\log\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 : \beta_j = 0\ (j\in\{1,\ldots,p\}\backslash S)\right\} \\ &+ n(\log(2\pi/n)+1) + 2\|\boldsymbol{\beta}^*\|_0 + 2,\end{aligned} \tag{3}$$

ただし, $\boldsymbol{\beta}^*$ は (3) の第一項の最小解である.

(3) を用いることで, 線形回帰における AIC 最小化は, 混合整数非線形計画問題と呼ばれる次の最適化問題に定式化できる.

$$\min_{\boldsymbol{\beta},\boldsymbol{z}}\left\{ n\log\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + 2\sum_{j=0}^{p} z_j : \begin{array}{l} z_j = 0 \Rightarrow \beta_j = 0\ (j=0,\ldots,p) \\ \beta_j \in \mathbb{R}\ (j=0,\ldots,p) \\ z_j \in \{0,1\}\ (j=0,\ldots,p)\end{array}\right\}. \tag{4}$$

問題 (4) を効率良く解く手法を 3 節で提案する.

3 提案する手法

この節では分枝限定法 [5] を用いた問題 (4) を効率良く手法を提案する. 分枝限定法は, 問題 (4) の部分問題の最小値の下界値と問題 (4) の最小値の上界値を更新する必要がある. 分枝限定法のフレームワークを提供している SCIP[1, 10, 14] に下界値と上界値の計算を実装することで, 問題 (4) を解くことができる.

問題 (4) の目的関数の第一項を

$$f(\boldsymbol{\beta}) = n\log\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2,$$

とおくと, 問題 (4) は次の問題で表される.

$$\min_{\boldsymbol{\beta},\boldsymbol{z}}\left\{ f(\boldsymbol{\beta}) + 2\sum_{j=0}^{p} z_j : \begin{array}{l} z_j = 0 \Rightarrow \beta_j = 0\ (j=0,\ldots,p) \\ \beta_j \in \mathbb{R},\ z_j \in \{0,1\}\ (j=0,\ldots,p)\end{array}\right\}. \tag{5}$$

まず, 問題 (5) の部分問題の最小値の下界値を計算する方法について述べる. 分枝限定法における分枝操作によって, 部分問題が生成される際, z_j $(j \in \{0,\ldots,p\})$ は 1 あるいは 0 に固定される. 生成された部分問題に対して, z_j $(j \in \{0,\ldots,p\})$ の添字に関して次の集合を定義する.

$$
\begin{aligned}
Z_1 &= \{j \in \{0,\ldots,p\} : z_j \text{ は } 1 \text{ に固定されている }\},\\
Z_0 &= \{j \in \{0,\ldots,p\} : z_j \text{ は } 0 \text{ に固定されている }\},\\
Z &= \{j \in \{0,\ldots,p\} : z_j \text{ はまだ固定されていない }\},
\end{aligned}
$$

ただし, $Z_1 \cup Z_0 \cup Z = \{0,\ldots,p\}$, $Z_1 \cap Z_0 = Z_1 \cap Z = Z_0 \cap Z = \emptyset$ である. 部分集合 $Z_1, Z_0, Z \subseteq \{0,\ldots,p\}$ に対して, 問題 (5) の部分問題 $Q(Z_1, Z_0, Z)$ は次のように表される.

$$
\min_{\boldsymbol{\beta},\boldsymbol{z}} \left\{ f(\boldsymbol{\beta}) + 2\sum_{j=0}^{p} z_j : \begin{array}{l} z_j = 1 \ (j \in Z_1),\ \beta_j = 0,\ z_j = 0 \ (j \in Z_0) \\ z_j = 0 \Rightarrow \beta_j = 0,\ z_j \in \{0,1\} \ (j \in Z) \\ \beta_j \in \mathbb{R} \ (j = 0,\ldots,p) \end{array} \right\}. \tag{6}
$$

部分問題 $Q(Z_1, Z_0, Z)$ の $z_j \in \{0,1\}$ $(j \in Z)$ の整数性を緩和した問題は次のように表される.

$$
\min_{\boldsymbol{\beta},\boldsymbol{z}} \left\{ f(\boldsymbol{\beta}) + 2\sum_{j=0}^{p} z_j : \begin{array}{l} z_j = 1 \ (j \in Z_1),\ \beta_j = 0,\ z_j = 0 \ (j \in Z_0) \\ z_j = 0 \Rightarrow \beta_j = 0,\ 0 \le z_j \le 1 \ (j \in Z) \\ \beta_j \in \mathbb{R} \ (j = 0,\ldots,p) \end{array} \right\}. \tag{7}
$$

問題 (7) の最小値は, 部分問題 $Q(Z_1, Z_0, Z)$ の最小値の下界値である. 次に問題 (7) から制約 $z_j = 0 \Rightarrow \beta_j = 0,\ 0 \le z_j \le 1$ $(j \in Z)$ と変数 z_j $(j \in \{1,\ldots,p\})$ を取り除いた以下の問題について考える.

$$
\min_{\boldsymbol{\beta}} \{ f(\boldsymbol{\beta}) + 2\#(Z_1) : \beta_j = 0 \ (j \in Z_0),\ \beta_j \in \mathbb{R} \ (j = 0,\ldots,p) \}. \tag{8}
$$

問題 (7) の任意の実行可能解 $(\bar{\boldsymbol{\beta}}, \bar{\boldsymbol{z}})$ に対して, 次の不等式が成立し,

$$
f(\bar{\boldsymbol{\beta}}) + 2\sum_{j=0}^{p} \bar{z}_j = f(\bar{\boldsymbol{\beta}}) + 2\left(\sum_{j \in Z} \bar{z}_j + \#(Z_1) \right) \ge f(\bar{\boldsymbol{\beta}}) + 2\#(Z_1),
$$

$\bar{\boldsymbol{\beta}}$ は (8) の実行可能解である. つまり, 問題 (8) の最小値は部分問題 $Q(Z_1, Z_0, Z)$ の最小値の下界値である. 問題 (7) の目的関数は非凸であるので, 問題 (7) を解くのは困難である. したがって, 本研究では, 問題 (8) を部分問題 $Q(Z_1, Z_0, Z)$ の最小値の下界値を求めるための緩和問題として扱う. なお, 問題 (8) を $R(Z_1, Z_0, Z)$ で表す. 問題 (8) は制約無し凸二次計画問題に変形することができるので, 線形方程式を解くことによって問題 (8) の最小値を得ることができる.

z_j を 1 に固定して生成される部分問題の緩和問題の最小値は, 生成元の部分問題の緩和問題の最小値を用いることで, 簡単に計算することができる. 部分問題 $Q(Z_1, Z_0, Z)$ に対して, $k \in Z$ を選択することで二つの部分問題 $Q(Z_1 \cup \{k\}, Z_0, Z\backslash\{k\})$, $Q(Z_1, Z_0 \cup \{k\}, Z\backslash\{k\})$ が生成される. z_k を 1 に固定して生成された部分問題 $Q(Z_1 \cup \{k\}, Z_0, Z\backslash\{k\})$ の緩和問題 $R(Z_1 \cup \{k\}, Z_0, Z\backslash\{k\})$ は次のよう表される.

$$
\min_{\boldsymbol{\beta}} \{ f(\boldsymbol{\beta}) + 2\#(Z_1 \cup \{k\}) : \beta_j = 0 \ (j \in Z_0),\ \beta_j \in \mathbb{R} \ (j = 0,\ldots,p) \}.
$$

したがって, 緩和問題 $R(Z_1, Z_0, Z)$ の最小値を θ^* とすれば, 緩和問題 $R(Z_1 \cup \{k\}, Z_0, Z \backslash \{k\})$ の最小値は $\theta^* + 2$ である.

次に, 問題 (5) の最小値の上界値を計算する方法について述べる. 緩和問題 $R(Z_1, Z_0, Z)$ の最小解を $\hat{\boldsymbol{\beta}} \in \mathbb{R}^p$, 最小値を $\hat{\theta}$ とする. $\hat{\boldsymbol{z}} \in \{0,1\}^p$ を

$$z_j = \begin{cases} 1 & (\text{if } \hat{\beta}_j \neq 0) \\ 0 & (\text{if } \hat{\beta}_j = 0) \end{cases} \quad (j = 0, \ldots, p),$$

と定義すれば, $(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{z}})$ は問題 (5) の実行可能解であり, 目的関数値は $\hat{\theta} + 2\#(\{j \in Z : \hat{z}_j = 1\})$ である. このようにして, 問題 (5) の最小値の上界値を得ることができる.

4 効率良く解くための工夫

4.1 ステップワイズ法

分枝限定法は, 解きたい問題の最小値の上界値とその部分問題の最小値の下界値を比べて, 部分問題が探索木から枝刈りできるかどうかをテストする. 上界値が下界値より小さければ枝刈りできるので, できるだけ小さい上界値を早い段階で得ることは, 分枝限定法の高速化を実現する. 実行可能解を得ることで, 上界値を計算することができるので, 実行可能解を得るための計算コストと枝刈りによる分枝限定法の高速化のトレードオフを考慮する必要がある.

分枝限定法を提供している SCIP[1, 10, 14] は, 混合整数 (非線形) 計画問題の実行可能解を見つけるための近似アルゴリズムが数多く実装されている. 本研究では, 問題 (5) のより良い実行可能解を得るために, ステップワイズ法を基にした近似アルゴリズムを SCIP に実装している. ステップワイズ法は, 計算が高速で, R などの既存の統計ソフトウェアに実装されている. AIC を評価基準とするステップワイズ法は, 例えば, 次のようなアルゴリズムである.

- [変数減少法]: 全て (あるいはいくつかの) 説明変数を採用した状態から, 最も AIC の値が改善されるようにモデルから説明変数を一つずつ取り除く. AIC の値が改善されなくなるとアルゴリズムは終了する.

- [変数増加法]: 一つも説明変数を採用していない (あるいはいくつかの説明変数を採用した) 状態から, 最も AIC の値が改善されるようにモデルに説明変数を一つずつ追加する. AIC の値が改善されなくなるとアルゴリズムは終了する.

どちらの手法も局所探索をしているとみなせるので, 得られる解が大域的最小解であるとは限らない.

提案する手法は, 部分問題に対して, 変数減少法 (変数増加法) を基にした 2 つの近似アルゴリズムを実行している. 具体的には, $j \in Z_1 \cup Z$ $(j \in Z_1)$ に対応する説明変数の候補を採用した状態から, $j \in Z$ に対応する説明変数の候補を取り除く (追加する) アルゴリズムを実装している. 実際は, 部分問題の親と子では, 実装している近似アルゴリズムによって

選ばれる説明変数は同じであることが多いので, 全ての部分問題に対してステップワイズ法を実行する必要は無い. 上界値の更新による枝刈りと計算コストのトレードオフを考慮して, どの部分問題に対してステップワイズ法を実行するかSCIPのパラメータで調整する必要がある.

4.2 分枝変数の選び方

問題 (5) に分枝限定法を適用する場合, z_j $(j \in Z)$ を0あるいは1に固定して部分問題を生成する. 固定する変数は分枝変数と呼ばれる. 分枝変数の選び方によって, 生成された部分問題が異なるので, 得られる下界値は異なる. 下界値が大きいほど, 強力な枝刈りのテストができるので, 分枝変数の選び方は, 分枝限定法の計算時間に大きく影響を与える. そのため, 様々な分枝変数の選び方が提案されている [1, 14]. 本研究では, 提案する手法にFull-strong-branching[1] と呼ばれる分枝変数の選び方を問題 (5) に合わせて効率よく実装している. さらに, 変数選択に現れる特徴的な構造を利用したMost-frequent-branchingを提案し, 実装している.

4.2.1 Full-strong-branching

Full-strong-branchingは, 全ての分枝変数の候補に対し, 生成される二つの部分問題の緩和問題の最小値を計算し, 評価関数を用いて, 緩和問題の最小値が大きい分枝変数を決定する. したがって, Full-strong-branchingを用いることで探索木上に生成される部分問題の最小値の下界値は大きくなるので, 枝刈りが発生しやすくなり, 探索木は小さくなる. Full-strong-branchingは, 得られる下界値が大きいという意味で最も効果的な分枝変数の選び方である. しかし, 分枝変数の候補全体の2倍の数の緩和問題を解く必要があるので, 通常, 計算コストが非常に大きい.

3節で述べたように, 提案する手法は, z_j $(j \in Z)$ が1に固定された部分問題の緩和問題の最小値は容易に計算でき, その最小値は, $j \in Z$ によらず一定である. このように緩和問題の計算コストを抑えることができるので, Full-strong-branchingを適用することで, 分枝限定法の高速化を期待できる. 5節の数値実験で, 提案する手法で実装しているFull-strong-branchingは, SCIPが問題 (5) に適用したSCIPに実装されている分枝変数の選び方よりも効果的であることを示す.

4.2.2 Most-frequent-branching

問題 (5) の目的関数が比較的小さい実行可能解において, 採用される頻度が多い説明変数の候補やほとんど採用されない説明変数の候補が存在する傾向がある. つまり, 採用される頻度が多い説明変数の候補を採用しなければ, 対応する実行可能解の目的関数値は大きくなることが期待できる. 本研究の提案する手法において, 分枝限定法の高速化の糸口の一つは, 小さい計算コストで, z_j $(j \in Z)$ を0に固定して生成される部分問題の最小値の下界値が大きくなるように分枝変数を決定することである. 採用される頻度が多い説明変数の候補に対応する z_j $(j \in Z)$ を分枝変数として決定すると, 生成される部分問題の実行可能領域に目的関数値の比較的小さい質の良い解が含まれないことを意味し, 緩和問題の最

小値が大きくなることが期待できる. 分枝限定法実行中に得られた実行可能解における採用される頻度の計算のコストは小さいため, 小いさい計算コストで分枝変数を決定することができる. 5節の数値実験では, 4.2.1節で述べたFull-strong-branchingと本節で述べたMost-frequent-branchingとSCIPが問題(5)に適用したSCIPに実装されている分枝変数の選び方を比較する.

5 数値実験

2.2節の線形回帰におけるAIC最小化の数値実験の結果を表1に示す. 数値実験のベンチマークデータは, UCI Machine Learning Repository [13] で公開されている線形回帰用のデータを使用した. なお, 値の単位や大きさの違いを考慮して, 全て説明変数の候補を平均0, 分散1, に正規化した.

表1の「n」はデータの数, 「p」は説明変数の候補の数を表す. 次の提案する手法と既存手法を比較する.

INF 本研究で提案する手法であり, SCIP 3.2.1 (Solving Constraint Integer Program[1, 10, 14]) を用いて問題(4)を解く. 分枝変数の選び方は, SCIPに実装されているInference-branchingを用いる.

MFB 本研究で提案する手法であり, SCIP 3.2.1を用いて問題(4)を解く. 分枝変数の選び方は, 4.2.2節で述べたMost-frequent-branchingを用いる.

FSB 本研究で提案する手法であり, SCIP 3.2.1を用いて問題(4)を解く. 分枝変数の選び方は, 4.2.1節で述べたFull-strong-branchingを用いる.

MISOCP[8] 混合整数二次錐計画問題を用いたAIC最小化の手法である. CPLEX 12.6.2[6]を用いる.

MIQP[3] 問題(4)は目的関数の第二項の値を$1,\ldots,p+1$に固定することで, $p+1$個の混合整数二次計画問題に分割することができる. $p+1$個の混合整数二次計画問題を$1,\ldots,p+1$の順にCPLEX 12.6.2を用いて解く.

「AIC」は得られたAICの値を表し, 値が最も小さい手法の数値を太字で記す. 「k」は採用された説明変数の数である. 「time(sec)」は計算時間を表し, 全ての手法において, 計算時間が5000秒を超えた場合は分枝限定法を打ち切った. したがって, その場合は, 記されているAICの値が最小値であるとは限らないが, 5000秒以内で得たAICの値の中で最小である値が記されている. 5000秒以内で解くことができた計算時間が最小である手法の数値を太字で記す. 「nodes」は, 分枝限定法において生成された部分問題の数を表す. 「gap」は最小値の上界値と下界値を用いて, 次のように定義される.

$$\text{gap}[\%] = \frac{\text{上界値} \ - \text{下界値}}{|\,\text{上界値}\,|} \times 100 \geq 0$$

分枝限定法が進むにつれgapは減少していき, 0になると, その時点での上界値を与えていた実行可能解に最適性が保証される. 5000秒以内で解くことができなかった場合, 計算を

打ち切ったときの gap を記している. 5000 秒以内で解くことができなかった手法がある場合, gap の値が最小である手法の数値を太字で記す. gap > 0 のとき, gap が小さくても得られた解に最適性は保証されないが, gap が小さい解は質の良い解であるという解釈ができる.

表 1 から, 以下の結果が読み取れる.

- 5000 秒以内に解くことができたデータにおいて, 最も速くアルゴリズムが終了する手法は, 提案する手法の MFB あるいは FSB である. 5000 秒以内に解くことができなかったデータにおいて, MISOCP アプローチと MIQP アプローチに比べ, `automobile` を除けば, 提案する手法の方が得られた AIC の値が小さい.

- MISOCP アプローチは, 少ない数の部分問題で求解できているが, 計算時間が大きい. 緩和問題の二次錐計画問題を解く計算コストが大きいことが理由の一つとして考えられる.

- 5000 秒以内に解くことができたデータにおいて, 三つの提案する手法を比べると, MFB と FSB の生成された部分問題の数は INF より少ない. これは, SCIP に実装されている Inference-branching より提案する手法の分枝変数の選び方の方が, 探索木に対する枝刈りが働いていることを意味する. 最もサイズが大きいベンチマークデータである `crime` に対する INF, MFB, FSB の結果に着目すると, 5000 秒で生成された部分問題の数は FSB が最も少なく, gap は最も小さい. これは, `crime` に対して Full-strong-branching は, 計算コストが大きいが, 大きい下界値を得ていることを意味する.

6 おわりに

本研究では, 線形回帰における AIC 最小化を混合整数非線形計画問題として定式化し, 分枝限定法を基に効率良く解く手法を提案した. 定式化された問題 (5) を高速に解くために, (i) 親問題の緩和問題の最小値を用いた下界値の計算, (ii) 緩和問題の最小解から元問題の実行可能解を生成, (iii) ステップワイズ法を基にしたヒューリスティックの実装 (iv) 変数選択に現れる特徴的な傾向を利用した分枝変数の選び方などの様々な工夫を SCIP に実装した.

線形回帰における AIC 最小化に対して, 提案する手法は, 既存手法 (MISOCP アプローチ [8], MIQP アプローチ [3]) に比べて, 高いパフォーマンスを発揮することを表 1 で示した. 説明変数の候補の数が 32 以下であれば, 安価な計算機でも現実的時間で, 提案する手法は AIC 最小化を行うことができている.

一つ目の今後の課題は, より p あるいは n が大きいデータに対する並列計算を用いた実装である. ParaSCIP と FiberSCIP[12] は, 分枝限定法のための並列計算のフレームワークを提供している. 二つ目の課題は, ベイズ情報量規準などの既に提案されている AIC 以外の情報量規準への適用である. 情報量規準に対して, 問題 (5) の目的関数を適当に定めることで解くことができる.

データ名	n	p	手法	AIC	k	time(sec)	nodes	gap(%)
servo	167	19	INF	**258.35**	9	1.17	7577	0.00
			MFB	**258.35**	9	0.79	4705	0.00
			FSB	**258.35**	9	**0.41**	2261	0.00
			MISOCP	**258.35**	9	19.12	2249	0.00
			MIQP	**258.35**	9	2.27	13464	–
auto-mpg	392	25	INF	**332.88**	15	4.06	18959	0.00
			MFB	**332.88**	15	**1.76**	5723	0.00
			FSB	**332.88**	15	2.68	11586	0.00
			MISOCP	**332.88**	15	382.47	2033	0.00
			MIQP	**332.88**	15	22.22	42886	–
solarflareC	1066	26	INF	**2816.29**	9	53.33	166639	0.00
			MFB	**2816.29**	9	**10.49**	32261	0.00
			FSB	**2816.29**	9	23.13	79015	0.00
			MISOCP	**2816.29**	9	489.77	9517	0.0
			MIQP	**2816.29**	9	26.49	47030	–
breastcancer	194	32	INF	**508.40**	10	505.70	3851×10^3	**0.00**
			MFB	**508.40**	10	478.66	3422×10^3	**0.00**
			FSB	**508.40**	10	**90.21**	550×10^3	**0.00**
			MISOCP	**508.40**	10	>5000	40×10^3	2.82
			MIQP	**508.40**	10	420.44	1788×10^3	–
forestfires	517	63	INF	**1429.64**	12	>5000	7480×10^3	1.11
			MFB	**1429.64**	12	>5000	13179×10^3	**0.77**
			FSB	**1429.64**	12	>5000	9938×10^3	0.95
			MISOCP	1431.58	10	>5000	7006	7.24
			MIQP	1435.07	7	>5000	2766×10^3	–
automobile	159	65	INF	-60.29	32	>5000	32192×10^3	**12.30**
			MFB	-61.28	32	>5000	29785×10^3	13.95
			FSB	-61.59	33	>5000	15300×10^3	16.43
			MISOCP	**-64.39**	31	>5000	219×10^3	20.61
			MIQP	52.84	8	>5000	17351×10^3	–
crime	1993	100	INF	**3410.25**	50	>5000	10272×10^3	0.78
			MFB	**3410.25**	50	>5000	9753×10^3	0.52
			FSB	**3410.25**	50	>5000	1904×10^3	**0.50**
			MISOCP	3690.67	14	>5000	140	18.30
			MIQP	3646.35	4	>5000	132×10^3	–

表 1: 提案する手法 (INF, MFB, FSB) と MISOCP アプローチと MIQP アプローチの比較

参考文献

[1] T. Achterberg: "SCIP: solving constraint integer progrms", Math. Prog. Comp., 1, 1, 1–41, 2009.

[2] H. Akeike: "A new look at the statistical model identification", IEEE Trans. Autom. Control, 19, 6, 716–723, 1974.

[3] D. Bertsimas, A. King and R. Mazumder: "Best Subset Selection via a Modern Optimization Lenz" , Ann. Stat., 44, 2, 813 – 852, 2016.

[4] I. Guyon and A. Elisseeff: "An Introduction to Variable and Feature Selection", J. Mach. Learn. Res., 3, 1157–1182, 2003.

[5] 茨木俊秀: "最適化の数学", 共立講座 21 世紀の数学 13, 共立出版, 2011.

[6] IBM ILOG CPLEX Optimizer 12.6.2, IBM ILOG 2015.

[7] K. Kimura and H. Waki: "Minimization of Akaike's Information Criterion via Mixed Integer Nonlinear Program", Proceedings in the 5th International Congress on Mathematical Software, Vol. 9725, 292 – 300, Berlin, 2016.

[8] R. Miyashiro and Y. Takano: "Mixed integer second-order cone programming formulations for variable selection", Eur. J. Oper. Res., 247, 721 – 731, 2015.

[9] T. Sato, Y. Takano, R. Miyashiro and A. Yoshise: "Feature Subset Selection for Logistic Regression via Mixed Integer Optimization", Comput. Optim. Appl., 2015.

[10] SCIP: Soving Constraint Integer Programs, `http://scip.zib.de/`

[11] R Development Core Team: "R: A Language and Environment for Statistical Computing", R Foundation for Statistical Computing, Vienna, Austria, 2008, `http://www.R-project.org`

[12] Y. Shinano, T. Achterberg, T. Berthold, S. Heinz and T. Koch: "ParaSCIP – a parallel extension of SCIP", Competence in High Performance Computing 2010, editors: C. Bischof, H.-G. Hegering, W. E. Nagel and G. Wittum, 135 – 148, Springer, 2012.

[13] UCI Machine Learning Repository, `http://archive.isc.uci.edu/ml/`

[14] S. Vigerske and A. Gleixner: "SCIP: Global Optimization of Mixed-Integer Nonlinear Programs in Branch-and-Cut Framework", ZIB-Report 16-24, Zuse Institute Berlin, May 2016.