The 33rd Annual Conference of the Japanese Society for Artificial Intelligence, 2019

変数間作用を考慮した非負スパースモデルの正則化経路探索

Entire regularization path for sparse nonnegative interaction model

高柳 未来 *1 Mirai Takayanagi　田部井 靖生 *2 Yasuo Tabei　西郷 浩人 *3 Hiroto Saigo

*1*3九州大学 Kyushu University　*2理化学研究所革新知能統合研究センター RIKEN Center for Advanced Intelligence Project

Building sparse combinatorial model with non-negative constraint is essential in solving real-world problems such as in biology, where the target response is often formulated by additive linear combination of features.This paper presents a solution by combining itemset mining with non-negative least squares.

Our contribution is a proposal of novel bounds specifically designed for the feature search problem.

In synthetic dataset, the proposed method is demonstrated to run orders of magnitudes faster than a naive counterpart without employing tree pruning. We also empirically show that non-negativity constraints reduce the number of active features much less than that of LASSO, leading to significant speed-ups in pattern search. In experiments using HIV-1 drug resistance dataset, the proposed method successfully model the rapidly increasing drug resistance triggered by accumulation of mutations in HIV-1 genetic sequences. We also demonstrate the effectiveness of non-negativity constraints in suppressing false positive features.

1. はじめに

ここ数 10 年で, スパースモデルは計算生物学や信号処理, メディア処理に始まる多様な分野への応用に成功している. 特に,ℓ_1 正則化線形回帰,LASSO として知られるこの問題とその亜種は広く研究されている [Hastie et al., 2015]. LASSO の大いに賞賛に値する性質は, 問題の凸定式化と, その結果高速なソルバが利用可能になることだ. さらに, いくつかの仮定の元では解の保証が得られる.

この問題で興味深い疑問は,LASSO で変数間作用を見つけられるかどうかである. 変数に対して指数関数的に増加するため, 相互作用の変数選択は困難だ. ここ 10 年で, この解決策として多くのアルゴリズムが開発されてきたが, ほとんどは変数間作用の次数を制限したり, 主効果のみを考える, 強い仮定をもつものであった. 実際, ほとんどの手法の変数間作用は極端な 2,3 次に限られている [Wu et al., 2009]. しかし, この手法は真の次数が指定したものよりも高い場合, その変数間作用を見つけるのに失敗する. 後者の仮定は, 最初に変数間作用を考えずに変数選択を行い, 最初に見つかる変数 (主効果と呼ばれる) を続く相互作用発見のステップでふるいにかける. この手法は, 主効果以外の作用や, 主効果と主効果以外に渡る変数間作用を発見できないのは明らかだ [Lim and Hastie, 2015]. ディープラーニングやカーネル法は, 変数間作用を組み入れた高精度予測関数を作ることができる. しかし, これらのブラックボックスな手法では顕著な特徴を選び出すことは困難である. 決定木やランダムフォレストは, 重要な変数も変数間作用も見つけることができるが, 分枝点での選択はヒューリスティックによるものであり, 大域的最適解が保証されず, 選択された変数は LASSO のように 1 貫性のあるモデル選択が存在しない.

前述の仮定によらずに, この難解な問題に取り組む研究はいくつかしかない. [Saigo et al., 2007] は頻出アイテムセットマイニングとブースティングを組み合わせた手法を提案した. 彼らのモデルは本質的に, 非線形の特徴を用いた線形モデルであり, 特徴はアイテムセットの列挙アルゴリズムによって見つける. 彼らはサンプル数 640, 特徴数 348 のデータセットで次数制限のない回帰を数分で終わらせた, と報告している. しかし, さらなる拡張性については述べられていない.

この論文では,LASSO に非負制約を追加する. 非負線形回帰 (NLS) は回帰係数が常に正の値をとると前もってわかっている場合に有効である. 例えば計算機生物学における質量分析法であり, これは観測されたスペクトラムが典型的な同位体のパターンに沿って復元できることから用いられる [Slawski1 et al., 2012]. 上に述べた予測問題は通常の線形回帰 (OLS) でも解くことは可能だということは, ふれておく価値がある.

非負の線形回帰を解くにあたって, 正則化パラメータの選択が重要な問題である. 標準的なグリッドサーチでは, 我々の問題の実行時間を増爆発的に加させた, なぜなら各正則化パラメータの値について変数の数の指数関数的な列挙を行う必要があるからだ. それゆえに, 我々は LASSO の正則化パス追跡アルゴリズムを採用した [Hastie et al., 2015]. それは有効な正則化パラメータがたどる経路を自動的に追跡することができ, グリッドの細かさの調整や, 不要な正則化パラメータの値を試行錯誤する必要がない. 我々の場合, 全ての範囲の正則化パラメータの値に対して, 全ての特徴の組み合わせを表示することができる. 明らかなことだが, スパース NLS の定式は LASSO を基にした他の手法のもの, 例えば SPP や iBoost のものとは異なり, 結果として有効な変数を見極めるための枝刈り条件も異なる.

この論文の貢献は 2 つある. 1 つは, 非負制約高次変数間作用モデルの正則化パス追跡のアルゴリズムであり, 我々の知る限り他の誰にも検討されたことがない. 組み合わせの特徴を探すにあたって, 効率的に有効な組を見極めつつできる限り不要な組み合わせを避けることができる, 新たな条件を提案する.

2. 背景

我々は n 個のサンプルからなり, それぞれが D 個の特徴と対応するラベルを持っており,$\{(\boldsymbol{z}_1, y_1), (\boldsymbol{z}_2, y_2), \ldots (\boldsymbol{z}_n, y_n)\}$, where $y \in \mathbb{R}^+$, $\boldsymbol{z} \in \{0,1\}^D$ と表される訓練データを扱う. さらに, 簡略化された計画行列 $\boldsymbol{Z} = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_n\}^\top$ を, 計画行列全体の $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots \boldsymbol{x}_n\}^\top$ と同様に利用し, アイテム

連絡先: afiveithree@gmail.com

Algorithm 1 Entire regularization path for nonnegative interaction model

Require: $\boldsymbol{Z}, \boldsymbol{y}$
$\boldsymbol{\beta} = \boldsymbol{0}, \mathcal{A} = \arg\max |\nabla L(\boldsymbol{\beta})|_j, \quad j \notin \mathcal{A}$ ▷ Initial Search
$\lambda = |\nabla L(\boldsymbol{\beta})|_{\mathcal{A}}, \boldsymbol{\gamma}_{\mathcal{A}} = \boldsymbol{1}, \boldsymbol{\gamma}_{j \notin \mathcal{A}} = 0$
while $\lambda > 0$ **do**
  $\tau_1 = \left(\frac{\beta_j}{\beta_j - \gamma_j}\right), \quad j \in \mathcal{A}$
  $\tau_2 = \frac{\lambda + (\nabla L(\beta))_j}{\lambda + (\nabla L(\beta))_j - (\nabla L(\gamma))_j}, \quad j \notin \mathcal{A}$ ▷ Main Search
  step length $\tau = \min\{\tau_1, \tau_2\}$
  if $\tau = \tau_1$ **then** remove the variable from active set $\mathcal{A}$.
  end if
  if $\tau = \tau_2$ **then** add the variable to active set $\mathcal{A}$.
  end if
  $\boldsymbol{\beta} \leftarrow (1-\tau)\boldsymbol{\beta} + \tau\boldsymbol{\gamma}$
  $\nabla L(\boldsymbol{\beta}) \leftarrow (1-\tau)\nabla L(\boldsymbol{\beta}) + \tau \nabla L(\boldsymbol{\gamma})$
  $\lambda \leftarrow (1-\tau)\lambda$
  Recalculate new direction $\boldsymbol{\gamma}_{\mathcal{A}} = -(\boldsymbol{X}_{\mathcal{A}}^\top \boldsymbol{X}_{\mathcal{A}})^{-1}\boldsymbol{1}_{\mathcal{A}}$
    $\boldsymbol{\gamma}_{j \notin \mathcal{A}} = 0$
end while

セットの存在と不在を次のように表す:

$$x_{i,t} = I(t \subseteq z_i), \forall t \in \mathcal{T}, \tag{1}$$

ここで $I(.)$ はアイテムセット (特徴の組)t がサンプル $\boldsymbol{x}$ に含まれるとき 1 を, それ以外で 0 を返す関数である. 特徴の組み合わせ空間の大きさを $p = |\mathcal{T}|$ とおけば, 全変数間作用を含むバイナリ行列を $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ と表せる. 我々のアルゴリズムでは $\boldsymbol{Z}$ に $\boldsymbol{X}$ の必要な部分行列のみが動的に追加される.

2.1 スパースな非負制約線形回帰

我々のモデルは非負制約付き最小 2 乗回帰 (NLS) に, ℓ_1 罰則項を導入した ℓ_1 正則化非負線形回帰 (L1NLS) すなわち非負制約付き LASSO であり, 応答ベクトル $\boldsymbol{y}$ を回帰係数ベクトル $\boldsymbol{\beta}$ を用いて, この問題は以下のように表す;

$$\min_{\boldsymbol{\beta} \succeq 0} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1, \tag{2}$$

ここで,λ は正則化パラメータであり,$\boldsymbol{\beta} \in \mathbb{R}^p$ はスパースな回帰係数ベクトルであり,$\boldsymbol{\beta} \succeq 0$ はその各要素が全て非負であることを示し,ℓ_1 罰則項によってその値はほぼ 0 である [Hastie et al., 2015]. ここで λ を 0 にすると, 非正則化 NLS の問題に 1 致する. 各データセットにおける最良の λ はデータに依存して決まり, グリッドサーチを用いた交差検証など, アダプティブ法に頼らなければならない [Hastie et al., 2015]. しかし我々の場合,λ ごとに列挙を行うのは高コストであり, この作業はすぐに手に負えなくなる. それゆえに, パス追跡アルゴリズムを有効利用し, 計算の負荷を可能な限り小さくしようと試みる. これは現在の λ から次の λ に移動して, 既知の特徴の情報から次の特徴を見つけることで実現される.

2.2 パス追跡アルゴリズム

[Rosset and Zhu, 2003] は 2 次の損失関数をもつ ell_1 正則化回帰は区分線形な解の経路を持ち, 通常の線形回帰と同じ時間的複雑性で効率よく計算できることを示した. 等式 2 から損失関数と罰則関数を分離し, $L(\boldsymbol{y}, \boldsymbol{X}\boldsymbol{\beta}) = \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2$ と $J(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\|_1$ を定義する. 我々の目標は回帰係数の外形

$$\hat{\boldsymbol{\beta}}(\lambda) = \arg\min_{\boldsymbol{\beta} \succeq 0} L(\boldsymbol{y}, \boldsymbol{X}\boldsymbol{\beta}) + \lambda J(\boldsymbol{\beta}), \tag{3}$$

を全ての λ について示すことである.

この問題に対する解が求められるのは $\hat{\boldsymbol{\beta}}(\lambda)$ が区分線形であるときだけであり, すなわち有限個の整列された λ の組, 例えば $\lambda_0 < \lambda_k \ldots < \lambda_\infty$ について, $\lambda_k \leq \lambda \leq \lambda_{k+1}$ であるときに, 次の式が成り立つときである. $\hat{\boldsymbol{\beta}}(\lambda_{k+1}) = \hat{\boldsymbol{\beta}}(\lambda_k) + (\lambda - \lambda_k)\gamma_k$ ここで $\gamma_k \in \mathbb{R}^p$ は λ_k の時のパスの方向であり,*equiangular vector* であると知られている.

[Rosset and Zhu, 2003] の 1 つ目の提案では, 勾配の等式 3 はテイラー展開を使って次式で与えられる.

$$\frac{\partial \hat{\boldsymbol{\beta}}(\lambda)}{\partial \lambda} = -\left[\nabla^2 L(\hat{\boldsymbol{\beta}}(\lambda)) + \lambda \nabla^2 J(\hat{\boldsymbol{\beta}}(\lambda))\right]^{-1} \nabla J(\hat{\boldsymbol{\beta}}(\lambda)), \tag{4}$$

この章で見るように, 非負 LASSO の式 2 は回帰係数が区分線形となる十分条件を満たし, 追跡し続けることが可能である.

より厳密には, 非負 LASSO の式 2 は 1 次のラグランジュの未定乗数法で次のように書き直せる.

$$\sum_i L(\beta) + \lambda \sum_j \beta_j - \sum_j \delta_j \beta_j, \tag{5}$$

ここで $\delta_j \geq 0$ はラグランジュ乗数で,$\boldsymbol{X}$ と $\boldsymbol{y}$ における L の依存は簡単のため省略した. Karush-Kuhn-Tucker(KKT) 条件より, 次の式を得る.

$$\nabla L(\beta)_j + \lambda - \delta_j = 0 \tag{6}$$

$$\delta_j \beta_j = 0 \tag{7}$$

λ は非負の値をとるので,λ に応じて次の条件が得られる.

$$\begin{cases} \lambda = 0 & (\nabla L(\beta))_j = 0, \quad \forall j, \\ \lambda > 0 & (\nabla L(\beta))_j = -\lambda < 0, \forall \{j : \beta_j > 0\} \end{cases} \tag{8}$$

もし $\lambda > 0$ なら,$(\nabla L(\boldsymbol{\beta}))_j$ は負になる. 他の興味深い例は λ が最大の時である. そのような場合,$\boldsymbol{\beta} = \boldsymbol{0}$ が式 2 の解となり, そのような λ は KKT 条件から見つけることができ,

$$\lambda_{max} = \min -(\nabla L(\boldsymbol{0}))_j = \max(\boldsymbol{X}^\top \boldsymbol{y})_j. \tag{9}$$

$\lambda > 0$ の場合を分析するために, 我々は変数 $\boldsymbol{\beta}$ から正の係数と 0 の係数を区別した. 有効変数の組を $\mathcal{A} = \{j : \hat{\beta}_j(\lambda) \neq 0\}$ とすると, 次の条件を満たす.

$$\begin{cases} if \quad j \notin \mathcal{A} \quad \nabla(L(\boldsymbol{\beta}))_j \leq -\lambda, & (10) \\ else if \quad j \in \mathcal{A} \quad \nabla(L(\boldsymbol{\beta}))_j = -\lambda. & (11) \end{cases}$$

上の条件は KKT 条件に基づき, 式 5 が最適である限り満たされる. その上, 変数が有効変数に含まれるのかについての情報がわかれば, 正則化パスの進む方向を計算できる;

$$\frac{\partial \hat{\beta}(\lambda)_{\mathcal{A}}}{\partial \lambda} = -(\boldsymbol{X}_{\mathcal{A}}^\top \boldsymbol{X}_{\mathcal{A}})^{-1} \boldsymbol{1}_{\mathcal{A}} \left(= \boldsymbol{\gamma}_{\mathcal{A}}\right), \tag{12}$$

ここで $\boldsymbol{X}_{\mathcal{A}}$ と $\boldsymbol{1}_{\mathcal{A}}$ はそれぞれ有効変数の組 $\mathcal{A}$ に限定された計画行列と, ベクトルを表現している.

我々の更新規則は $\boldsymbol{\beta}$ と $\boldsymbol{\gamma}$ のアフィン結合で表され,

$$\boldsymbol{\beta}_j^* \leftarrow (1-\tau)\boldsymbol{\beta}_j + \tau \boldsymbol{\gamma}_j \tag{13}$$

ここで $0 < \tau < 1$ はパスの更新距離. 実行可能な範囲では $\boldsymbol{\gamma}_{\mathcal{A}}$ は線形関数であるため, 条件 10,11 のどちらかが侵されない限

り, 簡単に解のパスを辿り続けることができる. もし, 条件が侵されれば, 次のどちらかが起きて方向が変化する;
変数の削除: この場合,$\boldsymbol{\gamma}_{\mathcal{A}}$ が同じ方向に進み続けると,$\nabla(L(\boldsymbol{\beta})_j) = -\lambda$ の条件が侵される. 転換点では,β_j^* が 0 になる. これが起きるまで更新の距離は $((1-\tau)\boldsymbol{\beta}+\tau\boldsymbol{\gamma})_j = 0$ を解いて τ を求めることで得られる.

$$\tau = \left(\frac{\beta_j}{\beta_j - \gamma_j}\right). \tag{14}$$

変数の追加: この場合,$\boldsymbol{\gamma}_{\mathcal{A}}$ が同じ方向に進み続けると,$\nabla(L(\boldsymbol{\beta}))_j \leq -\lambda$ の条件が侵される. 解は $1-\tau : \tau$ により $\nabla L(\boldsymbol{\beta})_j = -\lambda$ と $\nabla L(\boldsymbol{\gamma})_j = 0$ の境界線で別れる. そのため, 更新の距離はこの場合 $\nabla L((1-\tau)\boldsymbol{\beta}+\tau\boldsymbol{\gamma})_j = -(1-\tau)\lambda$ を解いて τ を求めることで得られる.

$$\tau = \frac{\lambda + (\nabla L(\beta))_j}{\lambda + (\nabla L(\beta))_j - (\nabla L(\gamma))_j}. \tag{15}$$

どちらの更新の距離も計算され, 小さい距離が選択される. これらの事象の生起の後, 解の経路の方向 12 は再計算する必要がある. アルゴリズム全体はアルゴリズム 1 に示す.

3. 手法

アルゴリズム 1 を計画行列 $\boldsymbol{X}$ 全体に適用するにあたって,$\boldsymbol{X}$ を最初に訓練データセット $\boldsymbol{Z}$ から抽出してから, パス追跡アルゴリズムを動かすのは考えやすいが非効率であり, アイテム数 D やサンプル数 n の増加とともに, 手に負えなくなる. そのため我々は $\boldsymbol{Z}$ から $\boldsymbol{X}$ の必要な部分を動的に抽出するように工夫した. これは,ℓ_1 ノルムによって引き起こされるスパース性によって, 有効な変数の数はそうでない変数より非常に少ないはずであり, 有効な変数だけを保てるという考えによる. ゆえに, アルゴリズム 1 の 2,6 行目の非有効変数から有効変数への切り替えにだけ焦点を当てる. 下記の通り,1 つ目は最初の探索,2 つ目は主探索と呼び, どちらの問題にも効率の良い分枝限定アルゴリズムを提案する. まず,Closed Itemset Mining (CIM) アルゴリズムの LCM ([Uno et al., 2003]) によって定義される標準的な探索空間を採用した. アイテムセットを含むデータベースが与えられると,CIM は高頻度で現れるアイテムセットを同じ頻度の部分集合を無視しながら列挙し, 同じ点を通らずにアイテムセットに対応する点に到達することができる. しかし, 探索点の数は特徴の数に対して指数関数的に増加するため, 枝刈りは非常に重要である. 下記の通り, 我々は各サンプルの持つ目標変数を使った効率的な分枝限定法の条件を提案する.

探索木の中の点 t に到達したと仮定し,$(\nabla L(\boldsymbol{\beta}))_t = \sum_{i=1}^n X_{ti}C_{it}$ where $C_{it} = X_{it}\beta_t - y_i$ とおくと, 次の定理によってさらに下の探索木に進むべきか判定することができる.

Theorem 1 次の条件が満たされるとき,

$$\sum_{\{i|C_{it}\geq 0\}} X_{ti}C_{it} + \lambda < 0, \tag{16}$$

t より先の点には解が存在せず, 最適解を失うことなく安全に部分木を刈ることができる.

最初の枝刈り条件と同じく,2 番目の枝刈り条件は $(\nabla L(\boldsymbol{\gamma}))_t = \sum_{i=1}^n X_{ti}D_{it}$ where $D_{it} = X_{it}\gamma_t - y_i$ とおくことで作成でき,

Theorem 2 次の条件が満たされるとき,

$$\sum_{\{i|D_{it}\leq 0\}} X_{ti}D_{it} = 0,$$

t より先の点には解が存在せず, 最適解を失うことなく安全に部分木を刈ることができる.

Theorem 3 最初の枝刈り条件と同じように,3 番目の条件は $(\nabla L(\boldsymbol{\gamma}))_t^* = \sum_{\{i|D_{it}\leq 0\}} X_{ti}D_{it}$, と $(\nabla L(\boldsymbol{\beta}))_t^* = \sum_{\{i|C_{it}\geq 0\}} X_{ti}C_{it} + \lambda$, とおくことで作成でき, 点 t より先にありうる最大の τ を既知の最大 τ_j と比較して,

$$\frac{\lambda + (\nabla L(\boldsymbol{\beta}))_t^*}{\lambda + (\nabla L(\boldsymbol{\beta}))_t^* - (\nabla L(\boldsymbol{\gamma}))_t^*} \geq \max_{knownj} \tau_j,$$

が満たされる場合,t より先の点には解が存在せず, 最適解を失うことなく安全に部分木を刈ることができる.

3.1 Initial search

最初の探索はアルゴリズム 1 の 2 行目で呼ばれるが, 1 度しか呼ばれず, 些細な問題である. これは次の最大化問題である;

$$\max\left(\boldsymbol{X}^\top \boldsymbol{y}\right)_j.$$

$(\nabla L(\boldsymbol{\beta}))_t = \sum_{i=1}^n X_{it}y_i$ とおき, 現時点で発見されている最大値を $(\nabla L(\boldsymbol{\beta}))_t^*$ とおくと, 次の定理を得る.

Theorem 4 次の条件が満たされるとき,

$$(\nabla L(\boldsymbol{\beta}))_t^* \geq \sum_{i=1}^n X_{it}y_i \tag{17}$$

t より先の点にはこれ以上の $(\nabla L(\boldsymbol{\beta}))_t$ が存在せず, 最適解を失うことなく安全に部分木を刈ることができる.

4. 人工データにおける実験

本章では, 提案した枝刈り条件の効率を人工データセットを用いて実証する. 人工データは次のように作成した. まず, ベルヌーイ分布 ($q = 0.6$) から乱数を生成し, 計画行列 $X \in \{0,1\}^{n\times p}$ を作成する. その後,5 つの特徴を選び出し, その 2^5 個の組み合わせの中から 5 つを選び出す. 真の係数 $\boldsymbol{w}$ は 1 様分布 $\mathcal{U}_{[0,1]}$ から生成し, 応答ベクトルを $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w}$ で作成する. 全ての計算時間は 64-bit コンピュータ Intel(R) Xenon(R) E502697 Processor 2.70GHz で測定されている.

図 1 では, 提案した枝刈り条件を利用する場合としない場合の計算時間を比較している. この実験では, 我々は特徴の数を 50 個に固定し, サンプル数を $n \in \{50, 100, 200, 500, 1000\}$ の間で変化させた. ここで特徴の数 D を 50 に設定するということは, 組み合わせの探索空間のサイズを最大で 2^{50} に設定することに対応するのを留意してほしい. それゆえに, 探索木の大きさと空間全体の探索点を訪れるための時間は指数関数的な速さで増大する. 事実, 図 1 では, ナイーブな手法では n が 200 以上に設定された場合,24 時間以内に終了しなかった. 対照的に, 我々の効率的な枝刈りは n を 1000 までに設定する限り, 計算時間を少なく保つことに成功した.

サンプル数を 50 に固定し, 特徴数 D を $\{20, 50, 100\}$, すなわち探索空間 $p = \{2^{20}, 2^{50}, 2^{100}\}$ に設定することでもよく似た観測が得られた. 図 2 では, ナイーブな手法の探索空間と探索時間は高速で増加し,1 方で, 提案された枝刈り条件を利用すると, 探索空間と時間は桁単位で抑えられることがわかる.

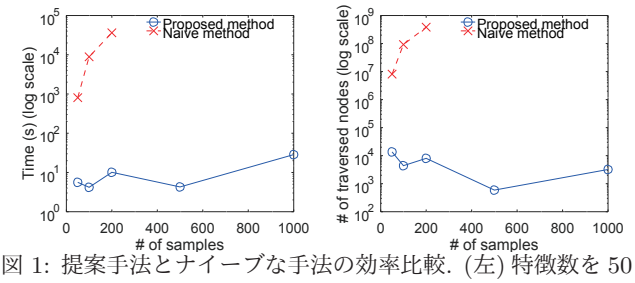

図 1: 提案手法とナイーブな手法の効率比較. (左) 特徴数を 50 に固定し, 秒単位の計算時間を示すサンプル数の関数として示す. (右) 同じように探索点の数をサンプル数の関数として示す.

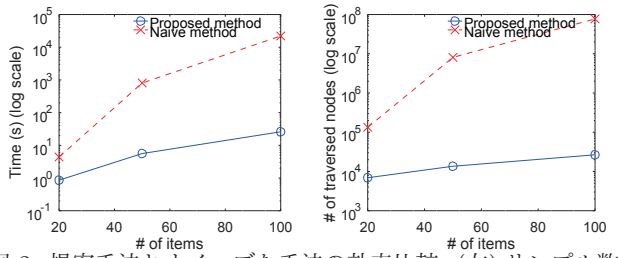

図 2: 提案手法とナイーブな手法の効率比較. (左) サンプル数を 50 に固定し, 秒単位の計算時間を特徴数の関数として示す. (右) 同じように探索点の数を特徴数の関数として示す.

5. 実データにおける実験

本章で, 我々は HIV-1 ウイルスの薬剤耐性データセット [Rhee et al., 2003] における提案手法の性能を実証する. データセットは市場で入手可能な薬剤に対する *in vitro* な感受性テストの結果を集めたものであり, 耐性値は野生型と比べた時の fold-change で記録されている. この記録は応答ベクトル $\boldsymbol{y}$ で構成されており, 遺伝子型は野生型の配列からの違いとして記録されている. 例えば, ある純粋培養された x が配列データベースにおける 1 番目, 六番目の 2 箇所に変異を持っており, それぞれ元となるアミノ基が Arginin,Cystein であることが判明した場合, 遺伝子配列は $\{1A, 6C\}$ の組として記録される.

表 1 において, 枝刈りによって得られる効率を実証する. 3. 章で導入した枝刈り条件に加えて, 最大パターンサイズ (アイテムセット内の特徴数) を説明のための枝刈り条件として採用した. 提案手法ではナイーブな手法に比べて 90% 以上の探索空間を削減することに成功していることがわかる. 加えて, 計算時間も約 90% 削減されていることがわかる.

最後に, アイテムセット LAR/LASSO と比較した提案手法の解釈可能性の改善を実証する.

2. 章ですでに議論したように,2 つの手法の違いは非負制約が課されているか, 否かという事実である. この検証のため, 両方のアルゴリズムを HIV-1 の AZT データセットに対して実行し,10 回目の繰り返しまでで得られた特徴を比較した. 表 2 は提案手法とアイテムセット LAR/LASSO により得られた特徴と, その係数を示し, 表は係数の絶対値で降順に並べた. どちらの手法でも, 最も影響力のある特徴の 1 つとして正の大きな重みを持つ $\{170L, 173E\}$ を特定している. しかし, アイテムセット LAR/LASSO は大きな負の係数を $\{139R\}$ にも割り当てている. しかし, 負の重みを持つ突然変異体は自然淘汰されるため, これはノイズであると考えられる.

謝辞

JSPS KAKENHI JP25700004 の後援に感謝します.

表 1: 提案手法と枝刈りの無いナイーブな手法における計算効率の比較

max. pat. size	Proposed method pruned	Proposed method time (s)	Naïve method pruned	Naïve method time (s)
2	96.8%	14.97	0%	90.69
3	98.2%	23.06	0%	726.81
4	98.7%	29.47	0%	422.89
5	98.7%	27.98	0%	340.56
6	99.0%	24.26	0%	383.66

表 2: 提案手法とアイテムセット LAR/LASSO で 10 回目の繰り返しの間に選ばれた特徴の組み合わせ

Absolute rank	Proposed method Coeff.	Proposed method Itemset	itemset LAR/LASSO Coeff.	itemset LAR/LASSO Itemset
1	108	60I	127	170L, 173E
2	101	170L, 173E	-85.1	139R
3	40.2	54I	71.2	60I, 173E
4	26.6	173E	45.7	60I
5	23.7	170L	-32.9	139R, 173E

参考文献

T. Hastie, R. Tibshirani, and M. Wainright. *Statistical Learning with Sparsity: The Lasso and Generalizations*. CRC Press, 2015.

M. Lim and T. Hastie. Learning interactions via hierarchical group-lasso regularization. *J Comput Graph Stat.*, 24(3): 627–654, 2015.

S. Y. Rhee, J. G. Matthew, R. Kantor, B. J. Betts, J. Ravela, and R. W. Shafer. Human immunodeficiency virus reverse transcriptase and protease sequence database. *NAR*, 31(1):298–303, 2003.

S. Rosset and J. Zhu. Piecewise linear regularized solution paths. Technical report, Stanford University, 2003. Technical Report HAL:ccsd-00020066.

H. Saigo, T. Uno, and K. Tsuda. Mining complex genotypic features for predicting HIV-1 drug resistance. *Bioinformatics*, 23(18):2455–2462, 2007.

M. Slawski1, R. Hussong, A. Tholey, T. Jakoby, B. Gregorius, A. Hildebrandt, and M. Hein. Isotope pattern deconvolution for peptide mass spectrometry by non-negative least squares/least absolute deviation template matching. *BMC Bioinformatics*, 13(291), 2012.

T. Uno, T. Asai, Y. Uchiyama, and H. Arimura. LCM: An efficient algorithm for enumerating frequent closed item sets. In B. Goethals and M. J. Zaki, editors, *Proceedings of the CEUR Workshop*, volume 90, New York, 2003. ACM Press.

T. T. Wu, Y. F. Chen, T. Hastie, E. M. Sobel, and K. Lange. Genome-wide association analysis by lasso penalized logistic regression. *Bioinformatics*, 25(6):714–721, 2009.