原著論文

Original Paper

Sparse Group Lassoのための高速なBlock Coordinate Descent

Fast Block Coordinate Descent for Sparse Group Lasso

井田 安俊
Yasutoshi Ida
日本電信電話株式会社 NTT ソフトウェアイノベー ションセンタ / 京都大学
NTT Software Innovation Center / Kyoto University
yasutoshi.ida@ieee.org

藤原 靖宏
Yasuhiro Fujiwara
日本電信電話株式会社 NTT コミュニケーション科学基礎研究所
NTT Communication Science Laboratories
yasuhiro.fujiwara.kh@hco.ntt.co.jp

鹿島 久嗣
Hisashi Kashima
京都大学
Kyoto University
kashima@i.kyoto-u.ac.jp

keywords: sparse group lasso, block coordinate descnent, feature selection, sparsity-inducing regularization

Summary

Sparse Group Lasso is a method of linear regression analysis that finds sparse parameters in terms of both feature groups and individual features. Block Coordinate Descent is a standard approach to obtain the parameters of Sparse Group Lasso, and iteratively updates the parameters for each parameter group. However, as an update of only one parameter group depends on all the parameter groups or data points, the computation cost is high when the number of the parameters or data points is large. This paper proposes a fast Block Coordinate Descent for Sparse Group Lasso. It efficiently skips the updates of the groups whose parameters must be zeros by using the parameters in one group. In addition, it preferentially updates parameters in a candidate group set, which contains groups whose parameters must not be zeros. Theoretically, our approach guarantees the same results as the original Block Coordinate Descent. Experiments show that our method enhances the efficiency of original method and achieves the same prediction error.

1. はじめに

Sparse Group Lasso (SGL)[Friedman 10, Simon 13] は特徴量にグループ構造を持つデータに対する線形回帰をベースとした特徴選択手法の一つである．特徴量にグループ構造が存在する場合，個々の特徴量を選択するだけでなく，予測値に関係の深い特徴グループを選択することも重要である．SGL は線形回帰における特徴量に対し，スパースな係数（パラメータ）を推定することで個々の特徴量と特徴グループを選択する．具体的には特徴レベルとグループレベルの 2 種類のスパース正則化項を用いることで効果的に前述の特徴選択を行う．こうした背景から SGL は特徴グループを持つ遺伝子発現データ [Nam 08, Roth 08] や天候データ [Ndiaye 16] の分析で広く使われる技術である．

SGL にてスパースなパラメータを推定するために Block Coordinate Descent (BCD) と呼ばれる手法が広く用いられる [Friedman 10, Simon 13]．BCD はそれぞれのグループに対応するパラメータを収束するまで順番に繰り返し更新するアルゴリズムである．その計算手続きは大きく二つに分けられる．まず BCD はある特徴グループに対応するパラメータベクトルがゼロベクトルになるか否かを判定する．これにより特徴グループレベルでの特徴選択を行うことができる．パラメータベクトルが非ゼロと判定された場合，BCD はそのグループのパラメータベクトルを更新する．BCD はこの手続きを全体のパラメータベクトルが収束するまですべてのグループに対し繰り返し行う．

SGL はグループ構造を持つデータの分析において有効な手法であるが，大規模なデータでは BCD の計算コストが高くなるという課題がある．主なボトルネックはパラメータベクトルがゼロベクトルになるか否かを判定する部分である．これは判定計算のためにパラメータ（特徴量）の次元数もしくはデータ数に比例した計算コストが必要となるためである．また，BCD による最適化の過程では，パラメータベクトルの多くがゼロベクトルとなる場合が多いにもかかわらず，上記のボトルネックとなる判定計算が全体のパラメータベクトルが収束するまで

すべてのグループに対して行われてしまう．これらの理由により，データの規模が大きくなるほど BCD の処理時間も大きくなってしまう．

本論文では SGL のための高速な BCD を提案する．メインとなるアイディアは，パラメータベクトルが必ずゼロとなるグループを小さい計算量で特定し，ボトルネックとなる判定計算をスキップするというものである．具体的には対象となるグループのパラメータベクトルの次元数に比例する計算量でこの特定を行う．グループ特徴量の次元数は全体の特徴量の次元数もしくはデータ数と比して小さい場合が多いため，本アイディアにより効率的にゼロベクトルとなるグループの計算をスキップすることができる．もう一つのアイディアは，パラメータベクトルが必ず非ゼロとなるグループの部分集合を含む集合（候補グループ集合）を抽出するというものである．候補グループ集合に対応するパラメータベクトルは予測に大きく貢献することが期待されるため，この集合内に対応するパラメータベクトルから優先的に更新を行うことにより，最適化の高効率化を期待できる [Fujiwara 16a, Fujiwara 16b, Ida 20]．提案手法は追加のハイパーパラメータを必要としないため，追加のチューニングコストも不要である．加えて，オリジナルの BCD と同じ目的関数の値への収束を保証できる．実験では提案手法が精度を劣化させずに処理時間を小さくすることを示す．

2. 事 前 準 備

2·1 Sparse Group Lasso

本節では Sparse Group Lasso (SGL) について説明する．SGL は線形回帰をベースとし，予測に関係する個々の特徴と特徴グループを見つける手法である．n をデータ数，p を特徴ベクトルの次元数とする．$y \in \mathbb{R}^n$ を n 個のデータに対応する回帰の正解値とし，G を特徴グループ数とする．$\boldsymbol{X} \in \mathbb{R}^{n \times p}$ を n 個のデータに対応する特徴量の行列とする．特徴グループの構造を明示的に記載すると，この特徴量の行列は $\boldsymbol{X} = [\boldsymbol{X}^{(1)}, \boldsymbol{X}^{(2)}, ..., \boldsymbol{X}^{(G)}]$ と表現される．ここで $\boldsymbol{X}^{(g)} \in \mathbb{R}^{n \times p_g}$ は p_g 次元の g 番目の特徴グループに対応する $\boldsymbol{X}$ の区分行列である．同様に，パラメータベクトル $\beta \in \mathbb{R}^p$ は $\beta = [\beta^{(1)\mathrm{T}}, \beta^{(2)\mathrm{T}}, ..., \beta^{(G)\mathrm{T}}]^{\mathrm{T}}$ と表現され，$\beta^{(g)} \in \mathcal{R}^{p_g}$ は g 番目のグループに対応するパラメータベクトル（回帰係数）である．したがって，SGL の線形回帰は $y = \boldsymbol{X}\beta = \boldsymbol{X}^{(1)}\beta^{(1)} + \cdots + \boldsymbol{X}^{(G)}\beta^{(G)}$ と記述できる．この SGL で推定されるパラメータベクトルは以下の問題を解くことで得られる．

$$\hat{\beta} = \underset{\beta \in \mathcal{R}^p}{\arg\min} \tfrac{1}{2}\|y - \textstyle\sum_{g=1}^{G} \boldsymbol{X}^{(g)}\beta^{(g)}\|_2^2 + (1-\alpha)\lambda \textstyle\sum_{g=1}^{G} \sqrt{p_g}\|\beta^{(g)}\|_2 + \alpha\lambda\|\beta\|_1. \quad (1)$$

ここで，$\alpha \in [0,1]$ と $\lambda \geq 0$ は正則化のハイパーパラメータを表す．α は l_1 ノルムと l_2 ノルムによる正則化のバランスを決め，λ は解のスパースさに影響する．式 (1) において，l_1 ノルムに対応する正則化はパラメータベクトルの各要素をスパースにするように働く．一方，l_2 ノルムに対応する正則化はパラメータのグループをスパースにするように働く．$\alpha = 0$ のとき，式 (1) は Group Lasso[Yuan 06] と一致する．$\alpha = 1$ のとき, この問題は Lasso[Tibshirani 96] となる. Sparse Group Lasso はこのように 2 種類の正則化を用いることで，特徴グループと個々の特徴の選択を同時に行う．

アルゴリズム 1 SGL のための BCD

```
1: 𝔸 = {1,...,G}, β ← 0;
2: repeat
3:     for each g ∈ 𝔸 do
4:         if ‖S(X^(g)T r_(−g), αλ)‖_2 ≤ √p_g (1−α)λ then
5:             β^(g) ← 0;
6:         else
7:             repeat
8:                 式 (2) による β^(g) の更新;
9:             until β^(g) が収束
10: until β が収束
```

2·2 Block Coordinate Descent

Block Coordinate Descent (BCD) は SGL の解 $\hat{\beta}$ を得るための標準的なアルゴリズムの一つである [Simon 13]．アルゴリズム 1 にその擬似コードを示す．SGL の BCD はグループに関する外側ループ（行番号 3–9）とグループ内に関する内側ループ（行番号 7–9）から構成される．外側ループはそれぞれのグループに対応するパラメータベクトル $\beta^{(g)}$ がゼロベクトルになるか否かを判定する（行番号 4）．$\beta^{(g)}$ がゼロベクトルでない場合，内側ループが $\beta^{(g)}$ の更新を行う（行番号 6–9）．この手続きは全体のパラメータベクトルが収束するまで繰り返し実行される（行番号 10）．

内側ループ（行番号 7–9）は $\beta^{(g)}$ が非ゼロベクトルと判定されたときに $\beta^{(g)}$ の更新を行う．その更新式は以下となる．

$$\beta^{(g)}_{\mathrm{new}} = \left(1 - \frac{t(1-\alpha)\lambda}{\|S(Z^{(g)}, t\alpha\lambda)\|_2}\right)_+ S(Z^{(g)}, t\alpha\lambda). \quad (2)$$

式 (2) では，$Z^{(g)} = \beta^{(g)} + t(\boldsymbol{X}^{(g)\mathrm{T}} r_{(-g)} - \boldsymbol{X}^{(g)\mathrm{T}}\boldsymbol{X}^{(g)}\beta^{(g)})$ と計算される．$t \geq 0$ はステップサイズである．$r_{(-g)}$ は partial residual と呼ばれ，次のように計算される．

$$r_{(-g)} = y - \textstyle\sum_{l=1, l\neq g}^{G} \boldsymbol{X}^{(l)}\beta^{(l)}. \quad (3)$$

$S(\cdot)$ はソフトしきい値 (soft-threshold operator) 処理を要素ごとに行う関数であり，例えば j 番目の要素は $S(z,\gamma)[j] = \mathrm{sign}(z[j])(|z[j]| - \gamma)_+$ のように計算される．$\beta^{(g)}$ は式 (2) によって収束するまで繰り返し更新される．$\beta^{(g)}$ がゼロベクトルと判定されている場合は式 (2) による更新処理

はスキップされる（行番号 4–5）．$Z^{(g)}$ の中の $X^{(g)\mathrm{T}}X^{(g)}$ はメインループ（行番号 2–10）に入る前に前計算可能であるため，式 (2) の時間計算量は $\mathcal{O}(p_g^2)$ となる．$X^{(g)\mathrm{T}}r_{(-g)}$ は次に説明するように，外側ループ（行番号 3–9）の中で計算される．

外側ループ（行番号 3–9）は次の判定式を用い，$\beta^{(g)}$ がゼロベクトルか否かを判定する．

$$\|S(X^{(g)\mathrm{T}}r_{(-g)}, \alpha\lambda)\|_2 \leq \sqrt{p_g}(1-\alpha)\lambda. \tag{4}$$

式 (4) は g 番目のグループのパラメータベクトルがゼロベクトルとなるための必要十分条件であり，式 (1) の l_2 ノルムの劣微分を用いた最適条件から導かれる [Friedman 10, Hastie 15, Simon 13, Sra 11, Yuan 06]．したがって式 (4) が満たされるとき，そのグループのパラメータベクトルはゼロベクトルとなり，内側ループの式 (2) はスキップされる（行番号 4–5）．式 (4) の中の $X^{(g)\mathrm{T}}r_{(-g)}$ は次のように計算される．

$$X^{(g)\mathrm{T}}r_{(-g)} = X^{(g)\mathrm{T}}y - X^{(g)\mathrm{T}}X\beta + X^{(g)\mathrm{T}}X^{(g)}\beta^{(g)}. \tag{5}$$

この式では $X^{(g)\mathrm{T}}y$，$X^{(g)\mathrm{T}}X$，$X^{(g)\mathrm{T}}X^{(g)}$ はメインループ（行番号 2–10）に入る前に前計算される．式 (5) の時間計算量は $\mathcal{O}(pp_g + p_g^2)$ となり式 (2) と比較して高い．加えて，この計算は式 (2) と異なりスキップされることはなく，アルゴリズムの収束まで繰り返し実行される．その結果，式 (4) の判定計算は $\mathcal{O}(pp_g + p_g^2)$ の時間計算量を毎イテレーションで要することになる．式 (5) の時間計算量は計算の仕方を変えることで $\mathcal{O}(np_g)$ とすることもできる [Huang 12]．しかし，式 (4) はどちらの時間計算量でも p もしくは n に依存することになる．つまり，式 (4) は特徴次元数，もしくはデータ数が大きい場合に高い計算コストがかかってしまう．

3. 提 案 手 法

本章では SGL の解を高速に得るための提案手法を説明する．最初にアルゴリズムの背後にあるアイディアを説明し，次にアルゴリズムを導出するための補題を紹介する．最後にアルゴリズムを擬似コードを用いて説明する．

3·1 アイディア

SGL の解を得るために使われる BCD は高い計算コストを要する．これは (i) 式 (4) が $\mathcal{O}(pp_g + p_g^2)$ または $\mathcal{O}(np_g)$ の時間計算量を要し，特徴次元数もしくはデータ数が大きい場合に計算量が増加してしまうためであり，また (ii) ほとんどのグループがゼロベクトルだったとしても BCD が式 (4) を収束するまで毎イテレーション実行してしまうためである．

提案手法のメインのアイディアは，式 (4) を近似し，必ずゼロベクトルになるグループを小さい計算量で特定することである．具体的には，式 (4) 内の $\|S(X^{(g)\mathrm{T}}r_{(-g)}, \alpha\lambda)\|_2$ を計算するのではなく，この値のある上界を計算する．この上界が $\sqrt{p_g}(1-\alpha)\lambda$ より小さければ，このグループのパラメータベクトルは必ずゼロベクトルになる．このアイディアによってゼロベクトルとなるグループを特定できれば，そのグループの計算を安全にスキップすることができる．式 (4) は $\mathcal{O}(pp_g + p_g^2)$ もしくは $\mathcal{O}(np_g)$ を要するが，この上界は $\mathcal{O}(p_g)$ で計算でき，時間計算量を削減することができる．

もう一つのアイディアはパラメータベクトルが必ず非ゼロベクトルになるグループの部分集合を含む候補グループ集合を抽出することである．候補グループ集合に対応するパラメータベクトルは予測に大きく寄与すると考えられる [Fujiwara 16a, Fujiwara 16b, Ida 20]．したがって，候補グループ集合内のグループから優先的にパラメータベクトルを更新することで効果的に最適化が行われることを期待できる．加えて，この候補グループ集合の構築は $\mathcal{O}(G)$ で行うことができるため，オーバーヘッドは比較的小さいと言える．

3·2 上 界

本節では式 (4) の $\|S(X^{(g)\mathrm{T}}r_{(-g)}, \alpha\lambda)\|_2$ の上界を用いて，必ずゼロベクトルをとるグループを特定する方法を説明する．タイトな上界を導出するため，外側ループに入る前のパラメータベクトルと現在のパラメータベクトルの差分を上界に取り入れる．パラメータベクトルが最適化の最中に収束するにつれ差分は小さくなり，またゼロベクトルが増えることによっても差分は小さくなるため，上界は最適化中に徐々にタイトになっていく．まず，ゼロベクトルとなるグループの特定に必要な定義を導入する．

定義 1 $U^{(g)}$ を以下のように定義する．

$$U^{(g)} = \|X^{(g)\mathrm{T}}\tilde{r}_{(-g)}\|_2 + \Lambda(g,g) + \textstyle\sum_{l=1}^{G}\Lambda(g,l). \tag{6}$$

ここで，$\tilde{r}_{(-g)}$ は外側ループに入る前の式 (3) の partial residual であり，$\Lambda(g,l) = \|\hat{K}^{(g)}[l]\|_2\|\beta^{(l)} - \tilde{\beta}^{(l)}\|_2$ である．$\hat{K}^{(g)}[l] \in \mathcal{R}^{p_g}$ の i 番目の要素は $\|K^{(g,l)}[i,:]\|_2$ で与えられる．この $\|K^{(g,l)}[i,:]\|_2$ は $K := X^{\mathrm{T}}X \in \mathcal{R}^{p\times p}$ の区分行列 $K^{(g,l)} \in \mathcal{R}^{p_g\times p_l}$ の i 番目の行ベクトルの l_2 ノルムである．$\tilde{\beta}^{(g)}$ は外側ループに入る前のパラメータベクトルである．

$\|X^{(g)\mathrm{T}}\tilde{r}_{(-g)}\|_2$ と $\|\hat{K}^{(g)}[\cdot]\|_2$ はそれぞれ外側ループ，メインループに入る前に前計算できることを記しておく．

次の補題はパラメータベクトルが必ずゼロベクトルになるグループに関して，この $U^{(g)}$ の性質を述べたものである．

補題 1（ゼロベクトルとなるグループ） $U^{(g)} \leq \sqrt{p_g}(1-\alpha)\lambda$ が成り立つとき，g 番目のグループのパラメータベクトル $\beta^{(g)}$ はゼロベクトルである．

補題 1 は $\|S(X^{(g)\mathrm{T}}r_{(-g)}, \alpha\lambda)\|_2$ の代わりに $U^{(g)}$ を使うことでゼロベクトルを持つグループを特定できることを

示している．これは付録 A の証明の過程に示すとおり，$U^{(g)}$ が式 (4) の $\|S(X^{(g)\mathrm{T}}r_{(-g)},\alpha\lambda)\|_2$ の上界となっているためである．この上界 $U^{(g)}$ の $\|S(X^{(g)\mathrm{T}}r_{(-g)},\alpha\lambda)\|_2$ に対するエラーバウンドは後に示す．

3·3 上界の逐次更新

上界 $U^{(g)}$ を使うことでゼロベクトルとなるグループを特定できることを前節で示した．しかし，式 (6) の上界 $U^{(g)}$ は $\|X^{(g)\mathrm{T}}\tilde{r}_{(-g)}\|_2$ と $\|\hat{K}^{(g)}[\cdot]\|_2$ を前計算したとしても $\mathcal{O}(p+p_g)$ の時間計算量を要する．通常の BCD が $\mathcal{O}(pp_g+p_g^2)$ もしくは $\mathcal{O}(np_g)$ であるため，このままだと上界の利用の恩恵は小さい．そこで，あるパラメータベクトルの更新が行われたときに式 (6) で $U^{(g)}$ を再計算するのではなく，$U^{(g)}$ を逐次的に計算し $U^{(g)'}$ へと更新することで計算量をさらに削減する．$U^{(g)'}$ は以下の定義で計算される．

定義 2（上界の逐次更新） $\beta^{(g)}$ が $\beta^{(g)'}$ へ更新されるとき，$U^{(g)'}$ を以下のように定義する．

$$U^{(g)'}=U^{(g)}-2\Lambda(g,g)+2\|\hat{K}^{(g)}[g]\|_2\|\beta^{(g)'}-\tilde{\beta}^{(g)}\|_2. \quad (7)$$

式 (7) は式 (6) から更新前の古い値である $2\Lambda(g,g)$ を引き，更新差分となる $2\|\hat{K}^{(g)}[g]\|_2\|\beta^{(g)'}-\tilde{\beta}^{(g)}\|_2$ を足しこんでいるため，式 (6) で再計算される $U^{(g)}$ と式 (7) で更新される $U^{(g)'}$ は同じ値となる．つまり，$U^{(g)'}$ も $\|S(X^{(g)\mathrm{T}}r_{(-g)},\alpha\lambda)\|_2$ の上界である．この逐次更新の計算コストは以下のようになる．

補題 2（上界の逐次更新の計算量） $\|X^{(g)\mathrm{T}}\tilde{r}_{(-g)}\|_2$ と $\|\hat{K}^{(g)}[\cdot]\|_2$ が前計算されている場合，g 番目のグループのパラメータベクトルが更新されたときの式 (7) の時間計算量は $\mathcal{O}(p_g)$ となる．

上記の補題は，逐次更新によって上界が $\mathcal{O}(p_g)$ で計算できることを示している．式 (4) は $\mathcal{O}(pp_g+p_g^2)$ (もしくは $\mathcal{O}(np_g)$)，式 (6) は $\mathcal{O}(p+p_g)$ を要するため，逐次更新の計算量は十分に小さいと言える．したがって，補題 1 と定義 2 を用いれば効率的にゼロベクトルとなるグループを特定できる．

3·4 候補グループ集合

本節ではパラメータベクトルが非ゼロベクトルになるグループの部分集合を含む候補グループ集合について説明する．候補グループ集合に含まれるグループから優先的にパラメータの更新を行うことにより，最適化が効率的に実行されることを期待する [Fujiwara 16a, Fujiwara 16b, Ida 20]．候補グループ集合を抽出する際には式 (4) の $\|S(X^{(g)\mathrm{T}}r_{(-g)},\alpha\lambda)\|_2$ を近似した次の基準を用いる．

定義 3（候補グループ集合を構築するための基準） g 番目のグループが候補グループ集合に含まれるか否かを判定する基準 $C^{(g)}$ を以下のように定義する．

$$C^{(g)}=\|X^{(g)\mathrm{T}}\tilde{r}_{(-g)}\|_2-\alpha\lambda\sqrt{p_g/2}. \quad (8)$$

ここで，$\tilde{r}_{(-g)}$ は外側ループに入る前の式 (3) で計算される partial residual である．

この基準 $C^{(g)}$ と上界 $U^{(g)}$ の $\|S(X^{(g)\mathrm{T}}r_{(-g)},\alpha\lambda)\|_2$ に対するエラーバウンドは次のように示すことができる．

補題 3（エラーバウンド） $\epsilon=\Lambda(g,g)+\sum_{l=1}^{G}\Lambda(g,l)+\alpha\lambda\sqrt{p_g/2}$ とすると，これは $|C^{(g)}-\|S(X^{(g)\mathrm{T}}r_{(-g)},\alpha\lambda)\|_2|\leq\epsilon$ となるような基準 $C^{(g)}$ の $\|S(X^{(g)\mathrm{T}}r_{(-g)},\alpha\lambda)\|_2$ に対するエラーバウンドである．また，上界 $U^{(g)}$ に対しては $|U^{(g)}-\|S(X^{(g)\mathrm{T}}r_{(-g)},\alpha\lambda)\|_2|\leq 2\epsilon$ が成り立つ．

上記の補題は基準 $C^{(g)}$ の $\|S(X^{(g)\mathrm{T}}r_{(-g)},\alpha\lambda)\|_2$ に対するエラーバウンドが上界 $U^{(g)}$ の $\|S(X^{(g)\mathrm{T}}r_{(-g)},\alpha\lambda)\|_2$ に対するエラーバウンドの半分となっていることを示しており，基準 $C^{(g)}$ が上界 $U^{(g)}$ よりもよく $\|S(X^{(g)\mathrm{T}}r_{(-g)},\alpha\lambda)\|_2$ を近似できる可能性があることを示している．この基準 $C^{(g)}$ を用いて候補グループ集合を以下のように定義する．

定義 4（候補グループ集合） 候補グループ集合 $\mathbb{C}$ を

$$\mathbb{C}=\{g\in\{1,...,G\}|C^{(g)}>\sqrt{p_g}(1-\alpha)\lambda\} \quad (9)$$

と定義する．

この候補グループ集合 $\mathbb{C}$ は次の性質を持つ．

補題 4（候補グループ集合） 候補グループ集合 $\mathbb{C}$ はパラメータベクトルが非ゼロベクトルになるグループの集合の部分集合を含む．

上記の補題における候補グループ集合は，パラメータベクトルが必ず非ゼロベクトルになるグループの部分集合と，非ゼロベクトルになりうるグループの両方を含むことに注意する．このパラメータベクトルが必ず非ゼロベクトルになるグループの部分集合とは，$L^{(g)}$ を $\|S(X^{(g)\mathrm{T}}r_{(-g)},\alpha\lambda)\|_2$ に対する下界としたときに $L^{(g)}>\sqrt{p_g}(1-\alpha)\lambda$ となるようなグループの集合である．詳細は補題 9 に記述する．また，候補グループ集合は空集合にもなりうることを記しておく．候補グループ集合 $\mathbb{C}$ の抽出にかかる計算コストは以下のようになる．

補題 5（候補グループ集合の計算コスト） $\|X^{(g)\mathrm{T}}\tilde{r}_{(-g)}\|_2$ が前計算されているとき，候補グループ集合 $\mathbb{C}$ は $\mathcal{O}(G)$ の時間計算量を要する．

上記の補題によると候補グループ集合はグループ数のオーダーで計算することができる．多くの大規模な問題ではグループ数よりも特徴次元数もしくはデータ数が大きくなるため，候補グループ集合の計算コストは比較的小さいと言える．

3·5 アルゴリズム

本節では前節までの定義や補題を利用した提案手法の説明を行う．アルゴリズム 2 にその擬似コードを記述する．提案手法は SGL の最適化で標準的なアプローチである BCD と sequential rule [Ghaoui 12] をベースにしている．Sequential rule は正則化のハイパーパラメータ λ を $\lambda_0>\lambda_1>...>\lambda_{Q-1}$ となるような系列 $(\lambda_q)_{q=0}^{Q-1}$ を用いてチューニングする手法である．パラメータベクトルは

それぞれの $(\lambda_q)_{q=0}^{Q-1}$ で系列の順番に最適化されるが，このときある λ の最適化では，その前の λ の最適化の結果のパラメータベクトルを初期値として用いる．

私たちのメインとなるアイディアは補題 1 を利用してパラメータベクトルが必ずゼロベクトルとなるグループを小さい計算量で特定し，大きな計算量を持つ式 (4) をスキップすることである．加えて，ある λ に対する最適化を行う際に，候補グループ集合を抽出する．候補グループ集合は補題 4 に示したとおり，非ゼロベクトルとなるパラメータベクトルを持つグループの部分集合を含んだ集合であり，このグループから優先的にパラメータベクトルを更新することで最適化が効率的に実行されることを期待する．候補グループ集合の抽出は補題 5 に示したように比較的小さい計算量で行うことができる．

アルゴリズム 2 では最初に上界の計算に用いる $\|\hat{K}^{(g)}[l]\|_2$ を前計算する（行番号 2–4）．候補グループ集合は sequential rule のループの中で最初に抽出される（行番号 6–10）．最初のイテレーションでは式 (9) を計算するが，次のイテレーションからは前のイテレーションの $\|X^{(g)\mathrm{T}}\tilde{r}_{(-g)}\|_2$ を式中で使い回すことで計算コストを削減できる．次に，候補グループ集合内のグループ上で BCD による最適化を行う（行番号 11–19）．その後，上界を用いた BCD のループに入る（行番号 20–36）．外側ループに入る前のパラメータベクトルを記録しておき（行番号 21），上界の計算に必要な $\|X^{(g)\mathrm{T}}\tilde{r}_{(-g)}\|_2$ を行番号 22–23 で前計算する．外側ループの中では g 番目のグループの上界 $U^{(g)}$ が式 (7) を用いて計算される（行番号 25）．ここで最初のイテレーションでは上界計算に式 (6) を用いることを記しておく．上界 $U^{(g)}$ がしきい値 $\sqrt{p_g}(1-\alpha)\lambda$ 以下であった場合，そのパラメータベクトルは補題 1 に従いゼロベクトルとなる（行番号 26–27）．上界がこの条件を満たさなかった場合，通常の BCD の計算が実行される（行番号 28–34）．そして上界計算に用いられる $\|\beta^{(g)}-\tilde{\beta}^{(g)}\|_2$ が行番号 35 で更新される．

このアルゴリズムの計算コストは以下のようになる．

定理 1（時間計算量） S と S' を，それぞれ補題 1 と式 (4) が使われたときにスキップされないグループの比率とする．すべてのグループのサイズを同一とみなし，p_g とする．t_m と t_f を BCD のメインループと内側ループの回数とすると，アルゴリズム 2 は $\mathcal{O}(G\{(Q+St_m)(pp_g+p_g^2)+p_g^2S't_mt_f\})$ もしくは $\mathcal{O}(G\{(Q+St_m)np_g+p_g^2S't_mt_f\})$ の時間計算量を要する．

定理 1 によると，補題 1 によって多くのグループがスキップされた場合，定理中の S が小さくなるためアルゴリズム 2 の時間計算量が小さくなる．つまり，最適化中にパラメータベクトルの多くがゼロベクトルとなる場合，多くのグループがスキップされると期待できるため，このような状況では提案手法による高速化を期待できる．

また，このアルゴリズムの収束に関しては以下のような性質がある．

定理 2（収束の性質） アルゴリズム 2 の正則化のハイパーパラメータが元の BCD と同一とし，さらに元の BCD が収束すると仮定する．すると，アルゴリズム 2 は元の BCD と同じ目的関数の値に収束する．

3·6 Overlapping Group/Graph Lasso への拡張

前節までの議論はグループが重ならないことを前提としているが，グループに重複があるユースケース（overlapping groups）も多く存在する．ここでグループに重複があるということは，異なるグループが同一の特徴量を共有している状態である．提案手法は overlap norm を用いた [Jacob 09] の実装方法を用いることで，このような場合にも比較的容易に手法を拡張することができる．

直感的には，[Jacob 09] の実装方法はそれぞれのグループに個別にパラメータベクトルを用意することで前述の重複問題に対処する．つまり，複数のグループが同一の要素を共有している特徴行列を $X\in\mathbb{R}^{n\times p}$ としたとき，共有されている要素を重複を許して $\tilde{X}\in\mathbb{R}^{n\times\sum_{g=1}^{G}p_g}$ と陽に表現する．重複がない場合は $\sum_{g=1}^{G}p_g=p$ となる．この特徴行列 $\tilde{X}$ に対応するようにパラメータベクトル $v^{(g)}$ を各グループ g に用意すると，線形回帰は $y=\sum_{g=1}^{G}\tilde{X}^{(g)}v^{(g)}$ と表せる．このとき overlap norm[Jacob 09] を用いると，グループの正則化項を $\lambda\sum_{g=1}^{G}\sqrt{p_g}\|v^{(g)}\|_2$ と表すことができる．したがって，グループに重複がある場合は以下の問題を解くことになる．

$$\begin{aligned}&\underset{v\in\mathbb{R}^{\sum_{g=1}^{G}p_g}}{\arg\min}\ \tfrac{1}{2}\|y-\textstyle\sum_{g=1}^{G}\tilde{X}^{(g)}v^{(g)}\|_2^2\\&+(1-\alpha)\lambda\textstyle\sum_{g=1}^{G}\sqrt{p_g}\|v^{(g)}\|_2+\alpha\lambda\|v\|_1.\end{aligned}\tag{10}$$

ここで $v\in\mathbb{R}^{\sum_{g=1}^{G}p_g}$ は $v=[v^{(1)\mathrm{T}},v^{(2)\mathrm{T}},...,v^{(G)\mathrm{T}}]^{\mathrm{T}}$ を表す．この問題は式 (1) の問題のサイズを変更しただけとみなせるため，提案手法を適用することができる．元のパラメータベクトル $\hat{\beta}$ を求める場合は，$\hat{\beta}=0$ とした上で，非ゼロとなった $v^{(g)}$ の要素のみを $\hat{\beta}$ の対応する要素へ加算していくことで求める．

[Jacob 09] によると，式 (10) の定式化を用いれば変数間にグラフ構造が与えられるような問題（Graph Lasso）も解くことができる．例えば，変数間のグラフ構造が無向グラフ (I,E) で与えられたときに重要なエッジを見つける問題を考える．このとき正則化項を $\lambda\sum_{e\in E}\sqrt{2}\|v^{(e)}\|_2$，$v^{(e)}$ をエッジが存在するある 2 ノードに対応する 2 要素からなるベクトルとし，1 エッジを 1 グループとみなせば重要なエッジを選択することができる．上記はエッジをグループ構造とみなしたが，クリークやサブグラフをグループ構造とみなすこともできる．

アルゴリズム 2 Fast Sparse Group Lasso

$\mathbb{A} = \{1, ..., G\}, \beta \leftarrow 0, \tilde{\beta} \leftarrow 0$;
for each $g \in \mathbb{A}$ **do**
  for each $l \in \mathbb{A}$ **do**
    $\|\hat{K}^{(g)}[l]\|_2$ の計算;
for $q = 0$ to $Q-1$ **do**
  $\mathbb{C} = \emptyset$;
  for each $g \in \mathbb{A}$ **do**
    式 (8) による $C^{(g)}$ の計算;
    if $C^{(g)} > \sqrt{p_g}(1-\alpha)\lambda_q$ **then**
      $\mathbb{C}$ へ g を追加;
  repeat
    for each $g \in \mathbb{C}$ **do**
      if $\|S(X^{(g)\mathrm{T}} r_{(-g)}, \alpha\lambda_q)\|_2 \leq \sqrt{p_g}(1-\alpha)\lambda_q$ **then**
        $\beta^{(g)} \leftarrow 0$;
      else
        repeat
          式 (2) による $\beta^{(g)}$ の更新;
        until $\beta^{(g)}$ が収束
  until β が収束
  repeat
    $\tilde{\beta} \leftarrow \beta$;
    for each $g \in \mathbb{A}$ **do**
      $\|X^{(g)\mathrm{T}} \tilde{r}_{(-g)}\|_2$ の計算;
    for each $g \in \mathbb{A}$ **do**
      式 (7) による $U^{(g)}$ の計算;
      if $U^{(g)} \leq \sqrt{p_g}(1-\alpha)\lambda_q$ **then**
        $\beta^{(g)} \leftarrow 0$;
      else
        if $\|S(X^{(g)\mathrm{T}} r_{(-g)}, \alpha\lambda_q)\|_2 \leq \sqrt{p_g}(1-\alpha)\lambda_q$ **then**
          $\beta^{(g)} \leftarrow 0$;
        else
          repeat
            式 (2) による $\beta^{(g)}$ の更新;
          until $\beta^{(g)}$ が収束
        $\|\beta^{(g)} - \tilde{\beta}^{(g)}\|_2$ の更新;
  until β が収束

4. 関 連 研 究

スパース正則化付き最適化を高速化するための有名な手法群としてスクリーニングがある [Ghaoui 12]．スクリーニングでは最適化に入る前にゼロになるパラメータ（あるいは不要な特徴量）を削除する．パラメータ及びデータ行列の大きさを削減できるため，最適化アルゴリズムの計算コストも下げることができる．SGL に対する最新のスクリーニング手法として，双対ギャップを利用した GAP Safe rule [Ndiaye 15, Ndiaye 16, Ndiaye 17, Wang 14] がある．この双対ギャップは SGL の主問題と双対問題の値の差として計算される．GAP Safe rule では解が含まれる安全領域を双対ギャップを用いて定義し，この領域に含まれないパラメータをゼロとする．安全領域が小さいほどこのアプローチは多くのグループや特徴を削除することができ，最適化も高速に行うことができる．しかし，λ が小さいか，あるいは解から遠いパラメータを初期値とした場合，スクリーニングはあまりパラメータを削除できず，最適化を効率的に実行することができないとされている [Johnson 16]．この問題を解決するため，GAP Safe rule に dynamic safe rule [Bonnefoy 14, Bonnefoy 15] を組み合わせた手法が提案されている [Ndiaye 17] ．この dynamic GAP Safe rule は BCD の最適化の最中に繰り返し GAP Safe rule を適用することで，パラメータが解に近づくにつれて削除されるパラメータの数を増加させることができる．

5. 実 験

本実験では計算時間と訓練誤差・予測誤差を六つのデータセット (*abalone*，*cpusmall*，*boston*，*bodyfat*，*eunite2001*，*pyrim*) で評価する．本データセットは LIBSVM のサイト [*1] から入手した．これらのデータセットのデータ数はそれぞれ 4177，8192，506，252，336，74 となる．これらのデータにグループ構造を与えるため，これらのデータセットの特徴量から多項式特徴量を作成した [Pavlidis 01]．本実験では [Roth 08] に従い 2 次の多項式特徴量を作成した．一つのグループ内には多項式カーネルを用いた二つの特徴量の 2 次までの組み合わせが入っている．この結果，それぞれのデータセットのグループ数は 36，78，91，105，136，378 となった．またそれぞれの合計の特徴次元数は 176，408，481，560，736，2133 となった．

比較手法として元の BCD，GAP Safe rule[Ndiaye 16]，dynamic GAP Safe rule[Ndiaye 17] を用いた．[Ndiaye 15, Ndiaye 16, Ndiaye 17, Wang 14] に従い，すべての手法で λ は sequential rule を用いてチューニングされる．探索空間は $\lambda_q = \lambda_{\max} 10^{-\delta q/Q-1}$ で定義される Q 個の非増加の系列 $(\lambda_q)_{q=0}^{Q-1}$ となる．ここで先行研究 [Ndiaye 15, Ndiaye 16, Ndiaye 17, Wang 14] に従い $\delta = 4$，$Q = 100$ とした．公正な比較のため，$\lambda_{\max}$ は双対ノルムを用いて計算した [Ndiaye 16]．比較手法の GAP Safe rule は本設定により安全にパラメータを削除できる．Dynamic GAP Safe rule

[*1] https://www.csie.ntu.edu.tw/~cjlin/libsvm/

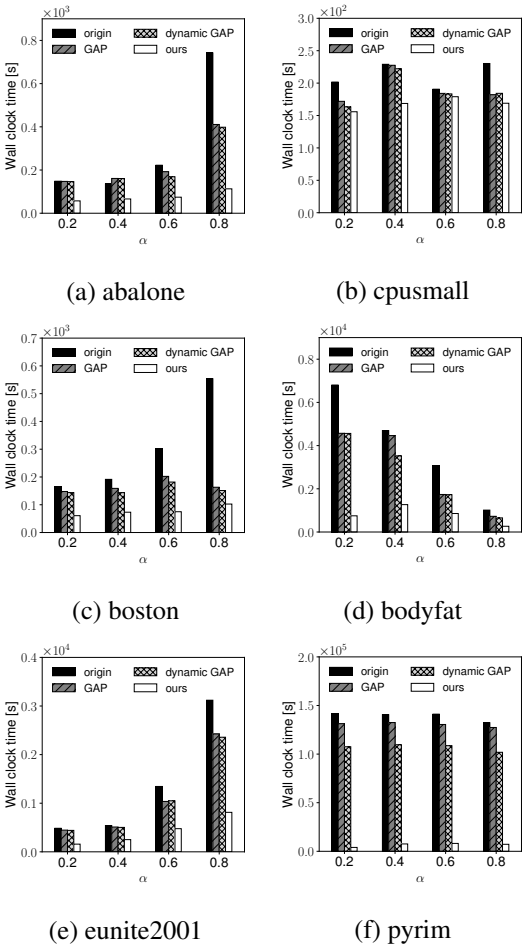


図 1 α 毎の計算時間.

においては双対ギャップの計算間隔を [Ndiaye 17] に従い 10 とした．もう一つチューニングが必要なハイパーパラメータ α においては，$\alpha \in [0.2, 0.4, 0.6, 0.8]$ を用いた．それぞれの手法は [Johnson 16, Johnson 17] に従い，それぞれの λ_q において，相対許容誤差 $\|\beta - \beta_{\text{new}}\|_2 / \|\beta_{\text{new}}\|_2$ が 10^{-5} 以下となったときに停止するものとした．すべての実験は 264GB メインメモリを持つ Linux 2.20 GHz Intel Xeon server 上で実行された．

5·1 計 算 時 間

本節ではそれぞれの $\alpha \in [0.2, 0.4, 0.6, 0.8]$ における sequential rule にかかる計算時間を評価する．図 1 に六つのデータセットのそれぞれの α についての計算時間を掲載する．本計算時間は前計算の時間も含んでいる．図の中で *origin*，*GAP*，*dynamic GAP*，*ours* はそれぞれ元の BCD，GAP Safe rule[Ndiaye 16]，dynamic GAP Safe rule[Ndiaye 17]，提案手法を表す．提案手法はすべてのデータセットと α で既存手法よりも高速であることがわかる．GAP Safe rule と dynamic GAP Safe rule は不要な特徴量に対応するパラメータを削除することで BCD を高速化している．しかし，これらの手法において削除するパラメータ数が少なく，BCD があまり高速化されない

表 1 ボトルネックとなる式 (4) の計算回数の比較

データセット	式 (4) の実行回数	
	origin	ours
abalone	1.141×10^5	$\mathbf{3.168 \times 10^3}$
cpusmall	2.105×10^5	$\mathbf{7.768 \times 10^4}$
boston	1.248×10^6	$\mathbf{9.998 \times 10^4}$
bodyfat	1.694×10^7	$\mathbf{2.403 \times 10^6}$
eunite2001	8.629×10^5	$\mathbf{2.052 \times 10^5}$
pyrim	7.667×10^7	$\mathbf{7.523 \times 10^6}$

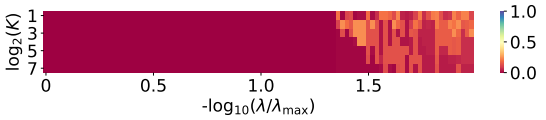


図 2 boston データセットの $\alpha = 0.2$ において式 (4) が実行された比率．横軸が sequential rule における各 λ，縦軸がイテレーション回数を表し，赤いほど比率が低い．

場合，双対ギャップの計算の分だけ計算時間が逆に大きくなるケースがあることを確認した．図 1(a) の $\alpha = 0.4$ の結果はそのケースに該当しており，元の BCD よりも計算時間が大きくなっている．

表 1 は BCD のボトルネックとなる式 (4) の実行回数を元の BCD と提案手法で比較したものである．この表から上界と候補グループ集合のアイディアにより，ボトルネックとなる計算の回数が削減されていることがわかる．また，この表における式 (4) の実行回数をより詳細に可視化したものが図 2 である．図 2 は boston データセットの $\alpha = 0.2$ において，式 (4) が実行された比率をヒートマップで表したものであり，横軸が sequential rule における各 λ，縦軸がイテレーション回数 K を表している．ヒートマップが赤い部分は式 (4) がよくスキップされていることを表す．λ が大きいほど，また BCD のイテレーションが進むほど式 (4) はよくスキップされており，この傾向は他のデータセットと α の場合も同様である．表 1 や図 2 は提案手法がボトルネックとなる計算をよくスキップしていることを示しており，これが図 1 の計算時間の短縮に繋がっていることを示唆している．

5·2 精 度

本節では提案手法の正確さを評価するため，訓練誤差と予測誤差を評価する．それぞれのデータセットの 50% を学習データ，50% をテストデータとし，両者の二乗誤差を評価した．表 2 が訓練誤差，表 3 が予測誤差の結果である．提案手法の誤差は元の BCD と一致していることがわかる．これは提案手法と元の BCD の目的関数の値が一致するという定理 2 を裏付ける結果である．

表 2 訓練誤差の比較.

データセット	訓練誤差	
	origin	ours
abalone	**2.209**	**2.209**
cpusmall	**7.862**	**7.862**
boston	**9.613**	**9.613**
bodyfat	$\mathbf{5.284 \times 10^{-3}}$	$\mathbf{5.284 \times 10^{-3}}$
eunite2001	$\mathbf{2.004 \times 10^{2}}$	$\mathbf{2.004 \times 10^{2}}$
pyrim	$\mathbf{4.568 \times 10^{-3}}$	$\mathbf{4.568 \times 10^{-3}}$

表 3 予測誤差の比較.

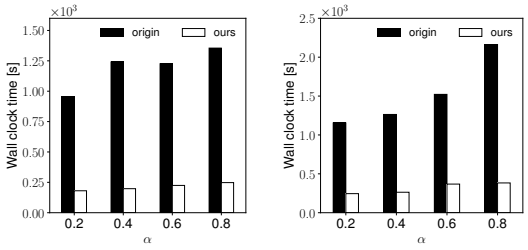
データセット	予測誤差	
	origin	ours
abalone	**2.232**	**2.232**
cpusmall	**7.886**	**7.886**
boston	**9.887**	**9.887**
bodyfat	$\mathbf{5.434 \times 10^{-3}}$	$\mathbf{5.434 \times 10^{-3}}$
eunite2001	$\mathbf{2.010 \times 10^{2}}$	$\mathbf{2.010 \times 10^{2}}$
pyrim	$\mathbf{4.615 \times 10^{-3}}$	$\mathbf{4.615 \times 10^{-3}}$

5·3 Overlapping Group/Graph Lasso

3·6 節では提案手法の Overlapping Group Lasso や Graph Lasso への拡張を述べた．本節では人工データを用いて，これらの拡張を行った場合でも提案手法が高速に動作することを見せる．

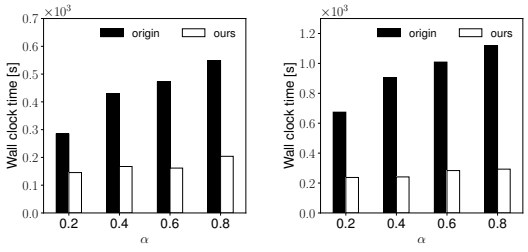
人工データの生成は [Ndiaye 16] と [Jacob 09] に従い以下のように行った．y は $y = X\beta + 0.01\epsilon$，$\epsilon \sim N(0, nI)$ として生成し，X は i 番目と j 番目の要素の相関が $0.5^{|i-j|}$ となるような平均 0 の多次元正規分布から生成する．β の各要素は一様乱数 $U \sim [0.5, 10]$，$\xi \sim [-1, 1]$ としたときに $\mathrm{sign}(\xi) \times U$ として生成する．Overlapping Group Lasso の実験では，$p = 1000, n = 100$，グループのサイズをそれぞれ 10 とし，グループ間の重複サイズを 2 と 5 とした 2 つのデータを生成した．例えば重複サイズが 2 の場合，グループのインデックスは $\{1, ..., 10\}, \{9, ..., 18\}, ...$ のようになる．Graph Lasso の実験では 3·6 節及び [Jacob 09] にあるように 1 エッジを 1 グループとみなすようデータの生成を行った．$p = 100, n = 100$ とし，グラフ構造は [Jacob 09] に従い鎖状のグラフとした．鎖の長さは 2 と 4 の 2 通りを生成した．その他，sequential rule のハイパーパラメータ設定などは前節までの実験設定に従った．

図 3 と図 4 にそれぞれ Overlapping Group Lasso と Graph Lasso の計算時間を示す．どちらの場合においても提案手法は高速であり，3·6 節の拡張を用いれば重複グループがある場合や変数にグラフ構造がある場合でも提案手法のアイディアを汎用的に使えることがわかる．



(a) 重複サイズ = 2. (b) 重複サイズ = 5.

図 3 Overlapping Group Lasso の α 毎の計算時間.



(a) 鎖の長さ = 2. (b) 鎖の長さ = 4.

図 4 Graph Lasso の α 毎の計算時間.

6. 結　　論

本論文では Sparse Group Lasso（SGL）のための高速な Block Coordinate Descent（BCD）を提案した．BCD の計算におけるメインのボトルネックは，あるグループのパラメータベクトルがゼロベクトルになるか否かを判定する計算である．この判定計算には特徴次元数もしくはデータ数に比例する計算コストがかかる．これに対し，提案手法はグループ内の特徴次元数の計算オーダーで必ずゼロベクトルになるグループを特定し，計算をスキップする．さらに，提案手法は必ず非ゼロベクトルになるグループの部分集合を含む候補グループ集合に対応するパラメータから優先的に更新を行うことで最適化の効率化を期待する．加えて，提案手法は元の BCD と同じ目的関数に収束することを現実的な仮定の元で保証する．実験では提案手法の精度が劣化せず，かつ既存手法と比較して高速であることを確認した．本論文ではグループが重複する場合（Overlapping Group Lasso）や特徴量にグラフ構造がある場合（Graph Lasso）への拡張も提案し，これらの設定においても提案手法が高速に動作することを実験で示した．

◇ 参　考　文　献 ◇

[Bonnefoy 14] Bonnefoy, A., Emiya, V., Ralaivola, L., and Gribonval, R.: A Dynamic Screening Principle for the Lasso, in *European Signal Processing Conference*, pp. 6–10, IEEE (2014)

[Bonnefoy 15] Bonnefoy, A., Emiya, V., Ralaivola, L., and Gribonval, R.: Dynamic Screening: Accelerating First-Order Algorithms for

the Lasso and Group-Lasso, *IEEE Trans. Signal Processing*, Vol. 63, No. 19, pp. 5121–5132 (2015)
[Foygel 10] Foygel, R. and Drton, M.: Exact Block-wise Optimization in Group Lasso and Sparse Group Lasso for Linear Regression, *arXiv preprint arXiv:1010.3320* (2010)
[Friedman 10] Friedman, J., Hastie, T., and Tibshirani, R.: A Note on The Group Lasso and a Sparse Group Lasso, *arXiv preprint arXiv:1001.0736* (2010)
[Fujiwara 16a] Fujiwara, Y., Ida, Y., Arai, J., Nishimura, M., and Iwamura, S.: Fast Algorithm for the Lasso based L1-Graph Construction, *Proc. VLDB Endow.*, Vol. 10, No. 3, p. 229–240 (2016)
[Fujiwara 16b] Fujiwara, Y., Ida, Y., Shiokawa, H., and Iwamura, S.: Fast Lasso Algorithm via Selective Coordinate Descent, in *Proceedings of AAAI Conference on Artificial Intelligence*, pp. 1561–1567 (2016)
[Ghaoui 12] Ghaoui, L. E., Viallon, V., and Rabbani, T.: Strong Rules for Discarding Predictors in Lasso-type Problems, *Pacific Journal of Optimization*, Vol. 8, No. 4, pp. 667–698 (2012)
[Hastie 15] Hastie, T., Tibshirani, R., and Wainwright, M.: *Statistical Learning with Sparsity: The Lasso and Generalizations*, Chapman & Hall/CRC (2015)
[Huang 12] Huang, J., Breheny, P., and Ma, S.: A Selective Review of Group Selection in High-Dimensional Models, *Statistical Science*, Vol. 27, No. 4, pp. 481–499 (2012)
[Ida 20] Ida, Y., Kanai, S., Fujiwara, Y., Iwata, T., Takeuchi, K., and Kashima, H.: Fast Deterministic CUR Matrix Decomposition with Accuracy Assurance, in *Proceedings of International Conference on Machine Learning*, pp. 4594–4603 (2020)
[Jacob 09] Jacob, L., Obozinski, G., and Vert, J.: Group Lasso with Overlap and Graph Lasso, in *Proceedings of International Conference on Machine Learning*, pp. 433–440 (2009)
[Johnson 16] Johnson, T. B. and Guestrin, C.: Unified Methods for Exploiting Piecewise Linear Structure in Convex Optimization, in *Advances in Neural Information Processing Systems*, pp. 4754–4762 (2016)
[Johnson 17] Johnson, T. B. and Guestrin, C.: StingyCD: Safely Avoiding Wasteful Updates in Coordinate Descent, in *Proceedings of International Conference on Machine Learning*, pp. 1752–1760 (2017)
[Nam 08] Nam, D. and Kim, S.-Y.: Gene-set Approach for Expression Pattern Analysis, *Brief. Bioinforma.*, Vol. 9, No. 3, pp. 189–197 (2008)
[Ndiaye 15] Ndiaye, E., Fercoq, O., Gramfort, A., and Salmon, J.: GAP Safe Screening Rules for Sparse Multi-task and Multi-class models, in *Advances in Neural Information Processing Systems*, pp. 811–819 (2015)
[Ndiaye 16] Ndiaye, E., Fercoq, O., Gramfort, A., and Salmon, J.: GAP Safe Screening Rules for Sparse-Group Lasso, in *Advances in Neural Information Processing Systems*, pp. 388–396 (2016)
[Ndiaye 17] Ndiaye, E., Fercoq, O., Gramfort, A., and Salmon, J.: Gap Safe Screening Rules for Sparsity Enforcing Penalties, *Journal of Machine Learning Research*, Vol. 18, No. 1, pp. 4671–4703 (2017)
[Pavlidis 01] Pavlidis, P., Weston, J., Cai, J., and Grundy, W. N.: Gene Functional Classification from Heterogeneous Data, in *Proceedings of the Fifth Annual International Conference on Computational Biology*, pp. 249–255, ACM (2001)
[Richtárik 14] Richtárik, P. and Takáč, M.: Iteration Complexity of Randomized Block-Coordinate Descent Methods for Minimizing a Composite Function, *Mathematical Programming*, Vol. 144, No. 1, pp. 1–38 (2014)
[Roth 08] Roth, V. and Fischer, B.: The Group-Lasso for Generalized Linear Models: Uniqueness of Solutions and Efficient Algorithms, in *Proceedings of International Conference on Machine Learning*, pp. 848–855 (2008)
[Simon 13] Simon, N., Friedman, J., Hastie, T., and Tibshirani, R.: A Sparse-Group Lasso, *Journal of Computational and Graphical Statistics*, Vol. 22, No. 2, pp. 231–245 (2013)
[Sra 11] Sra, S., Nowozin, S., and Wright, S. J.: *Optimization for Machine Learning*, The MIT Press (2011)
[Tibshirani 96] Tibshirani, R.: Regression Shrinkage and Selection via the Lasso, *Journal of the Royal Statistical Society*, Vol. 58, pp. 267–288 (1996)
[Wang 14] Wang, J. and Ye, J.: Two-Layer Feature Reduction for Sparse-Group Lasso via Decomposition of Convex Sets, in *Advances in Neural Information Processing Systems*, pp. 2132–2140 (2014)
[Yuan 06] Yuan, M. and Lin, Y.: Model Selection and Estimation in Regression with Grouped Variables, *Journal of the Royal Statistical Society*, Vol. 68, No. 1, pp. 49–67 (2006)

〔担当委員：小林 靖明〕

2019 年 11 月 15 日　受理

◇ 付　　録 ◇

本付録では本文に記載された補題と定理の証明を記載する．

A. 補題 1 の証明

補題 1 の証明を行う前に，次の二つの補題を証明する．

補題 6 g 番目のグループの $\|S(X^{(g)\mathrm{T}}r_{(-g)},\alpha\lambda)\|_2$ に対し，$\|X^{(g)\mathrm{T}}r_{(-g)}\|_2 \geq \|S(X^{(g)\mathrm{T}}r_{(-g)},\alpha\lambda)\|_2$ が成り立つ．

《証明》 g 番目のグループの $\|S(X^{(g)\mathrm{T}}r_{(-g)},\alpha\lambda)\|_2$ において $z = X^{(g)\mathrm{T}}r_{(-g)}$ とする．また，$\mathrm{sign}(\cdot)$ を引数のベクトルの各要素の符号を返す関数とし，$\mathrm{abs}(\cdot)$ を引数のベクトルの各要素の絶対値を返す関数とする．すると，以下の等式が成り立つ．

$$\begin{aligned}\|S(z,\alpha\lambda)\|_2 &= \|\mathrm{sign}(z)(\mathrm{abs}(z)-\alpha\lambda)_+\|_2 \\ &= \|(\mathrm{abs}(z)-\alpha\lambda)_+\|_2. \end{aligned} \tag{A.1}$$

上記において

$$\|(\mathrm{abs}(z)-\alpha\lambda)_+\|_2 \leq \|\mathrm{abs}(z)\|_2 = \|z\|_2 \tag{A.2}$$

が成り立つ．式 (A.1) と式 (A.2) より $\|S(z,\alpha\lambda)\|_2 \leq \|z\|_2$ が成り立つ．ここで $z = X^{(g)\mathrm{T}}r_{(-g)}$ より，補題 6 にある不等式を得る．□

次の補題は $U^{(g)}$ が $\|S(X^{(g)\mathrm{T}}r_{(-g)},\alpha\lambda)\|_2$ の上界になっていることを示す．

補題 7 g 番目のグループの $\|S(X^{(g)\mathrm{T}}r_{(-g)},\alpha\lambda)\|_2$ に対して，$U^{(g)} \geq \|S(X^{(g)\mathrm{T}}r_{(-g)},\alpha\lambda)\|_2$ が成り立つ．

《証明》 $K^{(g,:)} \in \mathcal{R}^{p_g \times p}$ を K の g 番目のグループに対応する区分行列とする．ここで簡略化のため $R^{(g)} := X^{(g)\mathrm{T}}r_{(-g)}$，$\tilde{R}^{(g)} := X^{(g)\mathrm{T}}\tilde{r}_{(-g)}$ とする．式 (5) より $R^{(g)}$ は

$$R^{(g)} = X^{(g)\mathrm{T}}y - K^{(g,:)}\beta + K^{(g,g)}\beta^{(g)}$$

と書ける．この式は差分 $\Delta\beta^{(g)} = \beta^{(g)} - \tilde{\beta}^{(g)}$ と $\Delta\beta = \beta - \tilde{\beta}$ を用いて

$$\begin{aligned}&X^{(g)\mathrm{T}}y - K^{(g,:)}\tilde{\beta} + K^{(g,g)}\tilde{\beta}^{(g)} + K^{(g,g)}\Delta\beta^{(g)} - K^{(g,:)}\Delta\beta \\ &= \tilde{R}^{(g)} + K^{(g,g)}\Delta\beta^{(g)} - \textstyle\sum_{l=1}^{G} K^{(g,l)}\Delta\beta^{(l)}\end{aligned}$$

と書ける．上記の式と三角不等式より，次の不等式を得る．

$$\|R^{(g)}\|_2 \leq \|\tilde{R}^{(g)}\|_2 + \|K^{(g,g)}\Delta\beta^{(g)}\|_2 + \textstyle\sum_{l=1}^{G}\|K^{(g,l)}\Delta\beta^{(l)}\|_2. \tag{A.3}$$

ここで式 (A.3) の 2 番目と 3 番目の項にある $K^{(g,l)}\Delta\beta^{(l)}$ の i 番目の要素は $K^{(g,l)}[i,:]\Delta\beta^{(l)}$ という内積で書ける．するとこの内積の絶対値の上界は，コーシー–シュワルツの不等式を用いることによって

$$|K^{(g,l)}[i,:]\Delta\beta^{(l)}| \leq \|K^{(g,l)}[i,:]\|_2\|\Delta\beta^{(l)}\|_2 \tag{A.4}$$

と書ける．加えて $\|\Delta\beta^{(l)}\|_2$ はスカラー値であるため，式 (A.3) の $\|K^{(g,l)}\Delta\beta^{(l)}\|_2$ に対し，式 (A.4) を用いて次の上界を得る．

$$\|K^{(g,l)}\Delta\beta^{(l)}\|_2 \leq \|\hat{K}^{(g)}[l]\|_2\|\Delta\beta^{(l)}\|_2 = \Lambda(g,l). \tag{A.5}$$

式 (A.3) と式 (A.5) より，$\|R^{(g)}\|_2$ の次の上界を得る．

$$\|R^{(g)}\|_2 \leq \|\tilde{R}^{(g)}\|_2 + \Lambda(g,g) + \textstyle\sum_{l=1}^{G}\Lambda(g,l) = U^{(g)}.$$

最後に，上記の上界と補題 6 を用いて，本補題の次の不等式を得る．

$$\|S(X^{(g)\mathrm{T}}r_{(-g)},\alpha\lambda)\|_2 \leq \|X^{(g)\mathrm{T}}r_{(-g)}\|_2 = \|R^{(g)}\|_2 \leq U^{(g)}.$$

□

上記の補題を用いて補題 1 の証明を行う．

《証明》 $U^{(g)} \leq \sqrt{p_g}(1-\alpha)\lambda$ が成り立つとき，補題 7 より $\|S(X^{(g)\mathrm{T}} r_{(-g)}, \alpha\lambda)\|_2 \leq U^{(g)} \leq \sqrt{p_g}(1-\alpha)\lambda$ が成り立つ．このとき g 番目のグループに対して $\|S(X^{(g)\mathrm{T}} r_{(-g)}, \alpha\lambda)\|_2 \leq \sqrt{p_g}(1-\alpha)\lambda$ が成り立つため，式 (4) より $\beta^{(g)}$ はゼロベクトルとなる． □

B. 補題 2 の証明

《証明》 式 (6) の計算により，$\|X^{(g)\mathrm{T}} \tilde{r}_{(-g)}\|_2$ と $\|\hat{K}^{(g)}[\cdot]\|_2$ はそれぞれ外側ループとメインループに入る前に前計算できる．したがって式 (6) 内の $\|X^{(g)\mathrm{T}} \tilde{r}_{(-g)}\|_2$，$\|\hat{K}^{(g)}[\cdot]\|_2$，$\Lambda(g,l)$ は $\mathcal{O}(1)$ の時間計算量で計算できる．$\beta^{(g)}$ が $\beta^{(g)'}$ へ更新されたとき，式 (6) の上界は式 (7) を用いて更新できる．$\|\beta^{(g)'} - \tilde{\beta}^{(g)}\|_2$ は $\mathcal{O}(p_g)$ の時間計算量を要するため，この更新計算は $\mathcal{O}(p_g)$ の時間計算量を要する．また $\Lambda(g,g)$ と $\|\hat{K}^{(g)}[g]\|_2$ は上述のように $\mathcal{O}(1)$ で計算できるため，g 番目のグループに対する式 (6) の上界は式 (7) を用いることで $\mathcal{O}(p_g)$ の時間計算量で計算できる． □

C. 補題 3 の証明

補題 3 を証明するために，次の定義と補題を導入する．

定義 5 $L^{(g)}$ を以下のように定義する．

$$L^{(g)} = \|X^{(g)\mathrm{T}} \tilde{r}_{(-g)}\|_2 - \Lambda(g,g) - \textstyle\sum_{l=1}^{G} \Lambda(g,l) - \alpha\lambda\sqrt{2p_g}. \quad \text{(A.6)}$$

ここで，$\tilde{r}_{(-g)}$ は外側ループに入る前の式 (3) の partial residual であり，$\Lambda(g,l) = \|\hat{K}^{(g)}[l]\|_2 \|\beta^{(l)} - \tilde{\beta}^{(l)}\|_2$ である．$\hat{K}^{(g)}[l] \in \mathcal{R}^{p_g}$ の i 番目の要素は $\|K^{(g,l)}[i,:]\|_2$ で与えられる．この $\|K^{(g,l)}[i,:]\|_2$ は $K := X^{\mathrm{T}} X \in \mathcal{R}^{p \times p}$ の区分行列 $K^{(g,l)} \in \mathcal{R}^{p_g \times p_l}$ の i 番目の行ベクトルの l_2 ノルムである．$\tilde{\beta}^{(g)}$ は外側ループに入る前のパラメータベクトルである．

この $L^{(g)}$ に関して次の補題が成り立つ．

補題 8 g 番目のグループの $\|S(X^{(g)\mathrm{T}} r_{(-g)}, \alpha\lambda)\|_2$ に対して，$L^{(g)} \leq \|S(X^{(g)\mathrm{T}} r_{(-g)}, \alpha\lambda)\|_2$ が成り立つ．

《証明》 ベクトル $I^{+}_{(g)} \in \{0,1\}^{p_g}$ の i 番目の要素は，$X^{(g)\mathrm{T}} r_{(-g)}$ の i 番目の要素の絶対値が $\alpha\lambda$ より大きいときに 1 をとり，その絶対値が $\alpha\lambda$ 以下のときに 0 をとるベクトルとする．また，abs(·) を引数のベクトルの各要素の絶対値を返す関数とする．すると $\|S(X^{(g)\mathrm{T}} r_{(-g)}, \alpha\lambda)\|_2$ はこれらを用いて次のように書き直すことができる．

$$\begin{aligned} &\|S(X^{(g)\mathrm{T}} r_{(-g)}, \alpha\lambda)\|_2 \\ &= \|\mathrm{abs}(X^{(g)\mathrm{T}} r_{(-g)}) - \alpha\lambda I^{+}_{(g)} - \mathrm{abs}(X^{(g)\mathrm{T}} r_{(-g)}) \odot (1 - I^{+}_{(g)})\|_2. \end{aligned} \quad \text{(A.7)}$$

ここで $\odot$ は要素積（アダマール積）である．上記の式と三角不等式を用いて次の不等式を得る．

$$\begin{aligned} &\|\mathrm{abs}(X^{(g)\mathrm{T}} r_{(-g)}) - \alpha\lambda I^{+}_{(g)} - \mathrm{abs}(X^{(g)\mathrm{T}} r_{(-g)}) \odot (1 - I^{+}_{(g)})\|_2 \\ &\geq \|X^{(g)\mathrm{T}} r_{(-g)}\|_2 - \|\alpha\lambda I^{+}_{(g)}\|_2 - \|X^{(g)\mathrm{T}} r_{(-g)} \odot (1 - I^{+}_{(g)})\|_2. \end{aligned} \quad \text{(A.8)}$$

ここで p^{+}_{g} を $I^{+}_{(g)}$ 内で 1 をとる要素の合計の数とすると，上記式内の $\|\alpha\lambda I^{+}_{(g)}\|_2$ は

$$\|\alpha\lambda I^{+}_{(g)}\|_2 = \alpha\lambda\sqrt{p^{+}_{g}} \quad \text{(A.9)}$$

と書ける．加えて，$X^{(g)\mathrm{T}} r_{(-g)} \odot (1 - I^{+}_{(g)})$ の各成分の絶対値は $\alpha\lambda$ 以下であるため，

$$\begin{aligned} &\|X^{(g)\mathrm{T}} r_{(-g)} \odot (1 - I^{+}_{(g)})\|_2 \leq \|\alpha\lambda(1 - I^{+}_{(g)})\|_2 \\ &= \alpha\lambda\sqrt{p_g - p^{+}_{g}} \end{aligned} \quad \text{(A.10)}$$

を得る．式 (A.7)，(A.8)，(A.9)，(A.10) より，以下の不等式を得る．

$$\begin{aligned} &\|S(X^{(g)\mathrm{T}} r_{(-g)}, \alpha\lambda)\|_2 \\ &\geq \|X^{(g)\mathrm{T}} r_{(-g)}\|_2 - \alpha\lambda\sqrt{p^{+}_{g}} - \alpha\lambda\sqrt{p_g - p^{+}_{g}} \\ &= \|X^{(g)\mathrm{T}} r_{(-g)}\|_2 - \alpha\lambda(\sqrt{p^{+}_{g}} + \sqrt{p_g - p^{+}_{g}}) \\ &\geq \|X^{(g)\mathrm{T}} r_{(-g)}\|_2 - \alpha\lambda\sqrt{2p_g}. \end{aligned} \quad \text{(A.11)}$$

上記の不等式では $\sqrt{p^{+}_{g}} + \sqrt{p_g - p^{+}_{g}} \leq 2\sqrt{p_g/2}$ の関係を用いた．次に，式 (A.11) の $\|X^{(g)\mathrm{T}} r_{(-g)}\|_2$ の項の下界を求める．上界の証明で用いた式 (A.3) と同じように，三角不等式を用いると $\|X^{(g)\mathrm{T}} r_{(-g)}\|_2$ の下界は

$$\begin{aligned} &\|X^{(g)\mathrm{T}} r_{(-g)}\|_2 = \|R^{(g)}\|_2 \\ &\geq \|\tilde{R}^{(g)}\|_2 - \|K^{(g,g)}\Delta\beta^{(g)}\|_2 - \textstyle\sum_{l=1}^{G} \|K^{(g,l)}\Delta\beta^{(l)}\|_2 \end{aligned} \quad \text{(A.12)}$$

と書ける．式 (A.12) と (A.5) より，以下の不等式を得る．

$$\|X^{(g)\mathrm{T}} r_{(-g)}\|_2 \geq \|\tilde{R}^{(g)}\|_2 - \Lambda(g,g) - \textstyle\sum_{l=1}^{G} \Lambda(g,l). \quad \text{(A.13)}$$

最後に，式 (A.11) と (A.13) を用いて補題にある次の不等式を得る．

$$\begin{aligned} &\|S(X^{(g)\mathrm{T}} r_{(-g)}, \alpha\lambda)\|_2 \\ &\geq \|X^{(g)\mathrm{T}} \tilde{r}_{(-g)}\|_2 - \Lambda(g,g) - \textstyle\sum_{l=1}^{G} \Lambda(g,l) - \alpha\lambda\sqrt{2p_g} = L^{(g)}. \end{aligned} \quad \text{(A.14)}$$

□

つまり，$L^{(g)}$ は $\|S(X^{(g)\mathrm{T}} r_{(-g)}, \alpha\lambda)\|_2$ に対する下界となっている．上記の補題 8 を利用することで，必ず非ゼロベクトルになるグループを次のように特定することができる．

補題 9（非ゼロベクトルとなるグループ） $L^{(g)} > \sqrt{p_g}(1-\alpha)\lambda$ が成り立つとき，g 番目のグループのパラメータベクトル $\beta^{(g)}$ は非ゼロベクトルである．

《証明》 補題 8 より，$L^{(g)} > \sqrt{p_g}(1-\alpha)\lambda$ が成り立つとき，$\|S(X^{(g)\mathrm{T}} r_{(-g)}, \alpha\lambda)\|_2 \geq L^{(g)} > \sqrt{p_g}(1-\alpha)\lambda$ が成り立つ．このとき $\|S(X^{(g)\mathrm{T}} r_{(-g)}, \alpha\lambda)\|_2 > \sqrt{p_g}(1-\alpha)\lambda$ が成り立つため，式 (4) より $\beta^{(g)}$ は非ゼロベクトルとなる． □

上記の補題 9 を用いて，補題 3 の証明を以下に記載する．

《証明》 補題 7 と 8 より

$$L^{(g)} \leq \|S(X^{(g)\mathrm{T}} r_{(-g)}, \alpha\lambda)\|_2 \leq U^{(g)} \quad \text{(A.15)}$$

が成り立つ．加えて，$C^{(g)}$ の定義より，

$$\frac{L^{(g)} + U^{(g)}}{2} = \|X^{(g)\mathrm{T}} \tilde{r}_{(-g)}\|_2 - \alpha\lambda\sqrt{p_g/2} = C^{(g)} \quad \text{(A.16)}$$

となり，

$$L^{(g)} \leq C^{(g)} \leq U^{(g)} \quad \text{(A.17)}$$

が成り立つ．上記より下界は $L^{(g)} = C^{(g)} - \Lambda(g,g) - \sum_{l=1}^{G} \Lambda(g,l) - \alpha\lambda\sqrt{p_g/2}$，上界は $U^{(g)} = C^{(g)} + \Lambda(g,g) + \sum_{l=1}^{G} \Lambda(g,l) + \alpha\lambda\sqrt{p_g/2}$ と書き直せる．したがって，$\epsilon = \Lambda(g,g) + \sum_{l=1}^{G} \Lambda(g,l) + \alpha\lambda\sqrt{p_g/2}$ とすると，$\|S(X^{(g)\mathrm{T}} r_{(-g)}, \alpha\lambda)\|_2$ の値は $C^{(g)} \pm \epsilon$ の間に存在することになる．これは言い換えれば

$$|C^{(g)} - \|S(X^{(g)\mathrm{T}} r_{(-g)}, \alpha\lambda)\|_2| \leq \epsilon \quad \text{(A.18)}$$

ということになり，補題中の不等式を得る．上界については，式 (A.15) より補題中の次の不等式を得る．

$$|U^{(g)} - \|S(X^{(g)\mathrm{T}} r_{(-g)}, \alpha\lambda)\|_2| \leq |U^{(g)} - L^{(g)}| = 2\epsilon. \quad \text{(A.19)}$$

□

D. 補題 4 の証明

《証明》 補題 9 より，$L^{(g)} > \sqrt{p_g}(1-\alpha)\lambda$ となるようなグループのパラメータベクトルは非ゼロベクトルとなる．また，定義 4 より，候補グループ集合は $C^{(g)} > \sqrt{p_g}(1-\alpha)\lambda$ となるようなグループの集合である．したがって式 (A.17) より，

$$L^{(g)} \leq C^{(g)} \quad \text{(A.20)}$$

が成り立つため，候補グループ集合は $C^{(g)} \geq L^{(g)} > \sqrt{p_g}(1-\alpha)\lambda$ となるようなグループを含む．つまり，候補グループ集合は補題 9 により特定される非ゼロベクトルを持つグループを含む． □

この補題 9 によって特定される非ゼロベクトルを持つグループの集合及び候補グループ集合は空集合にもなりうることを記しておく．

E. 補題 5 の証明

《証明》 $\|X^{(g)\mathrm{T}}\tilde{r}_{(-g)}\|_2$ が前計算されている場合，式 (8) は $\mathcal{O}(1)$ の時間計算量で計算できる．したがって，すべてのグループに対して式 (8) を計算すると $\mathcal{O}(G)$ の時間計算量を要する． □

F. 定理 1 の証明

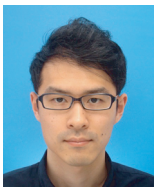
《証明》 アルゴリズム 2 において，まず $\mathcal{O}(p_g)$ を要する $\|\hat{K}^{(g)}[l]\|_2$ が前計算される．この計算をすべてのグループに対して行うため前グループで $\mathcal{O}(p)$ を要する．外側ループに入る前に，上界の計算のために $\|X^{(g)\mathrm{T}}\tilde{r}_{(-g)}\|_2$ が $\mathcal{O}(GQ(pp_g+p_g^2))$ もしくは $\mathcal{O}(GQnp_g)$ で前計算される．定義 2 より上界はパラメータベクトルが更新されたときに更新されるため，外側ループ 1 回分の時間計算量は補題 2 に従い $\mathcal{O}(GS'p_g)$ となる．上界によってそのグループがスキップされなかった場合，式 (4) が $\mathcal{O}(pp_g+p_g^2)$ もしくは $\mathcal{O}(np_g)$ で計算される．このスキップされない割合は S で表現されるため，外側ループ 1 回分の時間計算量は $\mathcal{O}(GS(pp_g+p_g^2))$ もしくは $\mathcal{O}(GSnp_g)$ となる．そのグループが式 (4) によってスキップされなかった場合は，パラメータベクトルが式 (2) によって更新される．式 (2) は前計算を用いると，式 (4) の $\|S(Z^{(g)},t\alpha\lambda)\|_2$ の計算に $\mathcal{O}(p_g)$，$Z^{(g)}$ の計算に $\mathcal{O}(p_g^2)$ を要するため，$\mathcal{O}(p_g^2)$ の時間計算量を要する．このスキップされなかった割合は S'，内側ループのイテレーション数は t_f で表されるため，時間計算量は $\mathcal{O}(GS't_fp_g^2)$ となる．上記の時間計算量をまとめ，BCD のメインループ数を t_m とすると，アルゴリズム 2 の時間計算量は $\mathcal{O}(G\{(Q+St_m)(pp_g+p_g^2)+p_g^2S't_mt_f\})$ もしくは $\mathcal{O}(G\{(Q+St_m)np_g+p_g^2S't_mt_f\})$ となる． □

G. 定理 2 の証明

《証明》 補題 1 より，$U^{(g)} \leq \sqrt{p_g}(1-\alpha)\lambda$ が成り立つグループは必ずパラメータベクトルがゼロベクトルとなるため，該当するグループの計算は安全に省略できる．その他の場合ではパラメータベクトルは非ゼロベクトルになりうるが，アルゴリズム 2 はこれらに対応するグループの計算は省略しない．つまり候補グループ集合に対応するグループを更新した後，アルゴリズム 2 は収束するまで安全に計算を省略しながらパラメータベクトルの更新を行う．したがって，通常の BCD が式 (1) 内の目的関数の最小値に収束すると仮定すると，アルゴリズム 2 も同じ最小値に収束する [Foygel 10, Hastie 15, Richtárik 14, Roth 08]. □

定理 2 は BCD が収束するという仮定のもと，「同じパラメータベクトル」ではなく「同じ目的関数の値」に収束することを保証する．これは候補グループ集合を優先的に更新することで元の BCD と更新順序が変わるためである．

著 者 紹 介



井田 安俊（正会員）

2014 年早稲田大学大学院先進理工学研究科電気・情報生命専攻修士課程修了．同年，日本電信電話株式会社入社．現在，同社 NTT ソフトウェアイノベーションセンタ研究員及び京都大学大学院情報学研究科知能情報学専攻博士課程に在学中．機械学習，データマイニングの研究開発に従事．

藤原 靖宏

2003 年早稲田大学大学院理工学研究科電気工学専攻修士課程修了．同年，日本電信電話株式会社入社．2011 年東京大学大学院情報理工学系研究科電子情報学専攻博士課程修了．2014 年ニューヨーク大学客員研究員．現在，日本電信電話株式会社 NTT コミュニケーション科学基礎研究所特別研究員．博士（情報理工学）．データマイニング，人工知能などの研究開発に従事．情報処理学会 2015 年度長尾真記念特別賞などを受賞．

鹿島 久嗣（正会員）

1999 年京都大学院工学研究科応用システム科学専攻修士課程修了．同年，日本アイ・ビー・エム株式会社入社．2007 年京都大学院情報学研究科知能情報学専攻博士後期課程修了．博士 (情報学)．2009 年より東京大学大学院情報理工学系研究科数理情報学専攻准教授を経て，2014 年より京都大学大学院情報学研究科知能情報学専攻教授．