1F4-4

選択的座標降下法による Lasso の高速化

藤原 靖宏[*1] 井田 安俊[*1] 塩川 浩昭[*2*1] 岩村 相哲[*1]
Yasuhiro Fujiwara Yasutoshi Ida Hiroaki Shiokawa Sotetsu Iwamura

[*1]NTT ソフトウェアイノベーションセンタ
NTT Software Innovation Center

[*2]筑波大学計算科学研究センター
Center for Computational Science, University of Tsukuba

The Lasso is an important regression approach and the coordinate descent algorithm is a standard approach to solve the Lasso. However, it has high computation cost. This paper proposes a fast approach to the Lasso. It achieves high efficiency by skipping unnecessary updates for the predictors whose weight is zero in the iterations. Experiments show that our approach can enhance the efficiency and the effectiveness of the Lasso.

1. はじめに

Lasso は高次元データに対する l_1 正則化付き最小 2 乗法の代表的な手法である．Lasso は 1990 年代の中頃に開発されたが [Tibshirani 96]，高い計算コストが必要なため 2000 年前半までは大きな注目を集めなかった．2007 年には高速な手法として座標降下法に基づく手法が提案された [Friedman 07]．さらに 2010 年及び 2012 年には座標降下法をさらに高速化する手法が提案された [Friedman 10, Tibshirani 12]．Lasso を計算するのに座標降下法に基づく手法は現在よく使われている．

しかし近年 Lasso が扱うデータは非常に大きなサイズになっている．Lasso が提案された 1990 年の中頃には高々 10 程度の予測変数が前立腺がんの解析に利用されていた．しかし最近の画像処理において予測変数の数は数千であり [Liu 14]，また話題抽出においては数万の予測変数が使われている [Kasiviswanathan 11]．本論文では Lasso において大規模のデータを高速に扱うために，座標降下法をより高速に行う手法を提案する．

2. 前準備

まず従来手法の説明を行う．回帰分析における予測変数の数を p とし，観測値の数を n とする．全てのベクトルは平均を 0 分散を 1 に正規化されているものとする．応答変数を $\mathbf{y} = (y[1], y[2], \ldots, y[n])^\top \in \mathcal{R}^n$ とする．また $\mathbf{X} \in \mathcal{R}^{n \times p}$ を p 個のベクトルからなる行列とし，$\mathbf{x}_i$ を行列 $\mathbf{X}$ における i 番目の列ベクトルとする．すなわち $\mathbf{x}_i$ は各予測変数に対応する．Lasso は行列 X において最小 2 乗誤差と l_1 正則化項の制限からなる以下の式を最小化する応答変数 y を予測する．

$$\min_{\mathbf{w} \in \mathcal{R}^p} \frac{1}{2n} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \lambda \|\mathbf{w}\|_1 \qquad (1)$$

ここで $\mathbf{w} = (w[1], w[2], \ldots, w[p])^\top$ は $w[i]$ が i 番目の予測変数 p_i に対応する係数のベクトルであり，また $\lambda > 0$ はチューニングパラメータと呼ばれる定数である．もし $\mathbf{X}$ が列フルランク行列であれば上記の最適化問題の解はただ一つになる．$p > n$ である場合は $\mathbf{X}$ は列フルランク行列にならない．アプリケーションにおいて最適な λ の値は異なるため，K をチューニングパラメータの数としたときに，チューニングパラメータを $\lambda_1 > \lambda_2 > \ldots > \lambda_K$ と変化させて最適化問題は解かれる．

座標降下法は他の係数はすでに更新したことを仮定し，係数 $w[i]$ に対して部分的に最適化を行う手法である．座標降下法は以下の式に基づいて係数を更新する．

$$\tilde{w}[i] \leftarrow S(z[i], \lambda) = \begin{cases} z[i] - \lambda & (z[i] > 0 \text{ and } |z[i]| > \lambda) \\ z[i] + \lambda & (z[i] < 0 \text{ and } |z[i]| > \lambda) \\ 0 & (|z[i]| \leq \lambda) \end{cases} \qquad (2)$$

式 (2) において $S(z[i], \lambda)$ は soft-thresholding operator と呼ばれる処理であり，また $z[i]$ は i 番目の予測変数に対応するパラメータとして以下の様に計算される．

$$z[i] = \frac{1}{n} \sum_{j=1}^{n} x[j, i](y[j] - \tilde{y}^{(i)}[j]) \qquad (3)$$

ここで $\tilde{y}^{(i)}[j] = \sum_{k \neq i} x[j, k]\tilde{w}[k]$ である．座標降下法による手法は係数を総当たり的に更新する．すなわち式 (2) を用いて収束するまで全ての予測変数の係数を繰り返し更新する．

Tibshirani らは更新において計算されるパラメータ $z[i]$ を以下の様に計算することで高速に求める手法を提案した．

$$z[i] = \tilde{w}[i] + \frac{1}{n} \left(\langle \mathbf{x}_i, \mathbf{y} \rangle - \sum_{j : |\tilde{w}[j]| > 0} \langle \mathbf{x}_i, \mathbf{x}_j \rangle \tilde{w}[j] \right) \qquad (4)$$

ここで $\langle \mathbf{x}_i, \mathbf{y} \rangle$ はベクトル $\mathbf{x}_i$ と $\mathbf{y}$ の内積であり，$\langle \mathbf{x}_i, \mathbf{y} \rangle = \sum_{j=1}^{n} x[j, i]y[j]$ と計算される．式 (3) と (4) は同じ計算結果となる．m を非零の係数を持つ予測変数の数とすると，係数を更新するのに必要は計算コストは式 (4) では $O(m)$ であり，式 (3) では $O(n)$ となる．Lasso では疎に予測変数が選択されるため，$m \ll n$ となる．そのため式 (4) を式 (3) の代わりに用いることによって，高速に係数を更新することが出来る．

Lasso をより高速に計算するために，Tibshirani らは以下の条件が成り立てはパラメータ λ_k において i 番目の予測変数を枝狩りする sequential strong rule を提案した．

$$\frac{1}{n} |\mathbf{x}_i^\top (\mathbf{y} - \mathbf{X}\tilde{\mathbf{w}}_{\lambda_{k-1}})| < 2\lambda_k - \lambda_{k-1} \qquad (5)$$

誤って係数が 0 にならない予測誤差が枝狩りされることがあるので，収束後に全ての予測変数に対して Karush-Kuhn-Tucker (KKT) 条件を満たすか確認を行う．KKT 条件とはもし $\tilde{w}_i = 0$ であれば $|\frac{1}{n} \mathbf{x}_i^\top (\mathbf{y} - \mathbf{X}\tilde{\mathbf{w}}_{\lambda_j})| < \lambda$ が成り立つという条件である．KKT 条件は式 (4) を用いることにより $O(m)$ のコストで計算できる．しかし大規模なデータを扱うため，より座標降下法を高速に行う必要がある．

3. 提案手法

従来手法は総当たり的に全ての予測変数に対して $z[i]$ を計算することで更新を行っていく．高速に Lasso の解を計算す

連絡先: 藤原靖宏，日本電信電話株式会社，〒 180-8585 東京都武蔵野市緑町 3-9-11，fujiwara.yasuhiro@lab.ntt.co.jp

るために，提案手法では全ての予測変数に対して更新を行わない．その代わり不必要な係数の更新をスキップする．具体的にはまず提案手法は必ず非零の係数となる予測変数だけを収束するまで更新し，そして非零の係数になり得る予測変数の更新を行う．繰り返し計算において更新を行う予測変数を決定するため，動的にパラメータ $z[i]$ の上限値と下限値を計算する．繰り返し計算の中で効果的に係数が 0 になる予測変数を枝狩りすることが出来るため，提案手法は従来手法より高速に Lasso の解を計算することができる．

3.1 上限値と下限値

提案手法は各繰り返し計算において参照ベクトルを設定し，各予測変数におけるパラメータ $z[i]$ の上限値と下限値を計算する．参照ベクトルは繰り返し計算に入る前の各予測変数における係数から構成される．$\tilde{\mathbf{w}}_r = (\tilde{w}_r[1], \tilde{w}_r[2], \ldots, \tilde{w}_r[p])^\top$ を参照ベクトルとしたときにパラメータ $z[i]$ の上限値は以下の様に定義される．

定義 1 $\overline{z}[i]$ をパラメータ $z[i]$ の上限値としたときに，$\overline{z}[i]$ は以下の様に与えられる．

$$\overline{z}[i] = \tilde{w}[i] - \tilde{w}_r[i] + \tfrac{1}{n}\|\mathbf{v}_i\|_2\|\tilde{\mathbf{w}} - \tilde{\mathbf{w}}_r\|_2 + z_r[i] \qquad (6)$$

式 *(6)* において $\mathbf{v}_i$ は j 番目の要素がベクトル $\mathbf{x}_i$ と $\mathbf{x}_j$ の内積 $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ である長さが p のベクトルとする．また $z_r[i]$ を参照ベクトル $\tilde{\mathbf{w}}_r$ で与えられるパラメータ $z[i]$ の値とする．

$$z_r[i] = \tilde{w}_r[i] + \tfrac{1}{n}\left(\langle \mathbf{x}_i, \mathbf{y}\rangle - \textstyle\sum_{j:|\tilde{w}_r[j]|>0} \langle \mathbf{x}_i, \mathbf{x}_j\rangle \tilde{w}_r[j]\right) \qquad (7)$$

ここで式 (6) において $\|\mathbf{v}_i\|_2$ と $z_r[i]$ は繰り返し計算に入る前に求めることが出来る．これはこれらが繰り返し計算において一定の値をとるためである．一方，$\|\tilde{\mathbf{w}} - \tilde{\mathbf{w}}_r\|_2$ に対して各繰り返し計算において必要となる計算コストは $O(p)$ となる．これは $\mathbf{w}$ 各繰り返し計算において更新される長さ p のベクトルだからである．同様に下限値は以下の様に定義される．

定義 2 $\underline{z}[i]$ をパラメータ $z[i]$ の下限値としたときに，$\underline{z}[i]$ は以下の式で計算される．

$$\underline{z}[i] = \tilde{w}[i] - \tilde{w}_r[i] - \tfrac{1}{n}\|\mathbf{v}_i\|_2\|\tilde{\mathbf{w}} - \tilde{\mathbf{w}}_r\|_2 + z_r[i] \qquad (8)$$

$\overline{z}[i]$ と $\underline{z}[i]$ がそれぞれ下限値と上限値となることを示すために以下の 2 つの補題を示す．

補題 1 i 番目の予測変数 p_i に対して，繰返し計算において $\overline{z}[i] \geq z[i]$ が成り立つ．

証明 $\mathbf{v}_i$ は j 番目の要素が $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ であるベクトルであるため，式 (4) と (7) から以下が成り立つ．

$$\begin{aligned} z[i] &= \tilde{w}[i] + \tfrac{1}{n}(\langle \mathbf{x}_i, \mathbf{y}\rangle - \langle \mathbf{v}_i, \tilde{\mathbf{w}}\rangle) \\ &= \tilde{w}_r[i] + \tilde{w}[i] - \tilde{w}_r[i] + \tfrac{1}{n}(\langle \mathbf{x}_i, \mathbf{y}\rangle - \langle \mathbf{v}_i, \tilde{\mathbf{w}}_r\rangle - \langle \mathbf{v}_i, \tilde{\mathbf{w}} - \tilde{\mathbf{w}}_r\rangle) \\ &= z_r[i] + \tilde{w}[i] - \tilde{w}_r[i] - \tfrac{1}{n}\langle \mathbf{v}_i, \tilde{\mathbf{w}} - \tilde{\mathbf{w}}_r\rangle \end{aligned}$$

コーシー・シュワルツの不等式から

$$-\tfrac{1}{n}\langle \mathbf{v}_i, \tilde{\mathbf{w}} - \tilde{\mathbf{w}}_r\rangle \leq \tfrac{1}{n}\|\mathbf{v}_i\|_2\|\tilde{\mathbf{w}} - \tilde{\mathbf{w}}_r\|_2$$

となるため，以下が成り立つ．

$$z[i] \leq \tilde{w}[i] - \tilde{w}_r[i] + \tfrac{1}{n}\|\mathbf{v}_i\|_2\|\tilde{\mathbf{w}} - \tilde{\mathbf{w}}_r\|_2 + z_r[i] = \overline{z}[i] \qquad \square$$

補題 2 繰返し計算において，i 番目の予測変数 p_i のパラメータ $z[i]$ に対して $\underline{z}[i] \leq z[i]$ が成り立つ．

証明 紙幅の都合により省略． $\square$

証明は省略するが，補題 2 は補題 1 と同様に式 (4) と (7) にコーシー・シュワルツの不等式を適用することで示せる．

式 (6) と (8) をそのまま用いて上限値と下限値を計算すると，$O(p)$ の計算コストが必要となる．これは $\|\tilde{\mathbf{w}} - \tilde{\mathbf{w}}_r\|_2$ を計算するのに $O(p)$ の計算コストが必要となるからである．しかし $\tilde{w}'[i]$ を更新を行う前の $\tilde{w}[i]$ の係数としたときに，$\|\tilde{\mathbf{w}} - \tilde{\mathbf{w}}_r\|_2$ は以下のように高速に更新することが出来る．

$$\|\tilde{\mathbf{w}} - \tilde{\mathbf{w}}_r\|_2 = \sqrt{\|\tilde{\mathbf{w}}' - \tilde{\mathbf{w}}_r\|_2^2 - (\tilde{w}'[i])^2 + (\tilde{w}[i])^2} \qquad (9)$$

式 (9) から $\|\tilde{\mathbf{w}} - \tilde{\mathbf{w}}_r\|_2$ は $O(1)$ で更新できるため，上限値と下限値の計算において以下が成り立つ．

補題 3 各予測変数に対して上限値と下限値は各繰返し計算において $O(1)$ で計算することが出来る．

証明 係数が更新されたとき $\|\tilde{\mathbf{w}} - \tilde{\mathbf{w}}_r\|_2$ は $O(1)$ で計算することができる．また繰返し計算に入る前に $\|\mathbf{v}_i\|_2$ と $z_r[i]$ は計算できる．そして $\tilde{w}[i] - \tilde{w}_r[i]$ は $O(1)$ の計算量で求めることが出来る．そのため式 (6) と (8) を用いて上限値と下限値は $O(1)$ の計算コストで求めることが出来る． $\square$

3.2 非零の係数を持つ予測変数

提案手法では上限値と下限値を用いて非零の係数を持つ予測変数を特定し更新を行う．提案手法は係数が 0 になる予測変数の更新を枝刈りし，高速に Lasso の解を求める．提案手法は以下に示す 2 つの係数についての性質を用いる．

補題 4 予測変数 p_i は $\underline{z}[i] > \lambda$ または $\overline{z}[i] < -\lambda$ が成り立てば必ず非零の係数を持つ．

証明 紙幅の都合により省略． $\square$

補題 5 予測変数 p_i はもしパラメータ $z[i]$ に対して $\overline{z}[i] > \lambda$ または $\underline{z}[i] < -\lambda$ が成り立てば非零の係数を持つ可能性がある．

証明 紙幅の都合により省略． $\square$

予測変数が補題 4 の条件を満たすとき，その予測変数は必ず補題 5 の条件を満たす．すなわち補題 4 による予測変数の集合は必ず補題 5 による予測変数の集合に含まれる．これはもし $\underline{z}[i] > \lambda$ であれば $\overline{z}[i] > \lambda$ であり，$\overline{z}[i] < -\lambda$ であれば $\underline{z}[i] < -\lambda$ であるからである．そのためもし予測変数において補題 4 より係数が非零となるとき，その予測変数は補題 5 より非零の係数を持つ可能性があるとされる．さらに予測変数が補題 5 の条件を満たさないとき，補題 5 よりその予測変数の係数は 0 となる．提案手法は補題 4 と 5 を用いて効果的に非零の係数を持つ予測変数を更新する．

3.3 アルゴリズム

Algorithm 1 に提案手法のアルゴリズムを示す．提案手法は Lasso を解く従来手法に基づいている．従来手法は sequential strong rule により不必要な予測変数を枝刈りし，式 (4) を用いて係数を更新する．提案手法は従来手法と同様に複数のチューニングパラメータを $\lambda_1 > \lambda_2 > \ldots > \lambda_K$ と設定し，それぞれのチューニングパラメータに対する Lasso の解を求める．Algorithm 1 において $\mathcal{U}$ を繰返し計算において更新を行う予測変数の集合とし，$\mathcal{P}$ を全ての予測変数の集合とし，$\mathcal{P}_k$ をチューニングパラメータ λ_k への解として非零の係数となる予測変数の集合とし，$\mathcal{S}_k$ をチューニングパラメータ λ_k における sequential strong rule で枝刈りされなかった予測変数の

集合とする．チューニングパラメータ λ_k と λ_{k-1} に対する解は似ているという知見に基づき，提案手法はチューニングパラメータ λ_k に対する係数の初期値を決定する．具体的には $w_{\lambda_k}[i]$ を λ_k における解の予測変数 p_i の係数としたとき，係数を初期値を以下のように設定する．

$$\tilde{w}[i] = \begin{cases} 0 & (k=1) \\ w_{\lambda_{k-1}}[i] & (k=2) \\ w_{\lambda_{k-1}}[i] + \Delta w_{\lambda_{k-1}}[i] & (k \geq 3) \end{cases} \quad (10)$$

ここで $\Delta w_{\lambda_{k-1}}[i] = w_{\lambda_{k-1}}[i] - w_{\lambda_{k-2}}[i]$ である．

Algorithm 1 において提案手法はまず $\lambda := \lambda_k$ とし，もし $k=1$ であれば $\mathcal{U} := \emptyset$ とし，そうでなければ従来手法と同様に $\mathcal{U} := \mathcal{P}_{k-1}$ とする (2 〜 6 行目)．係数の初期値は式 (10) を用いて計算する (7 行目)．パラメータ $z[i]$ の上限値と下限値を計算するのに用いる参照ベクトルは繰返し計算に入る前に設定する (10 〜 12 行目と 19〜21 行目)．繰返し計算においてまず提案手法は補題 4 における条件を用いて非零の係数を必ず持つ予測変数に対して更新を行う (13 〜 18 行目)．そして補題 5 を用いて非零の係数を持つ可能性ある予測変数に対して更新を行う (22 〜 29 行目)．収束後，$p_i \in \mathcal{S}_k$ であるような予測変数と $p_i \in \mathcal{P}$ であるような予測変数に対してそれぞれ KKT 条件を用いて係数が非零になる予測変数がないか確認を行う (30 〜 33 行目及び 33 〜 36 行目)．これらの処理は従来手法と同じである．

それぞれのチューニングパラメータに対して解を求めた後，事前に決めた基準に対して最適な解を設定する．計算コストにおいて提案手法は以下の性質を持つ．

定理 1 m_k をチューニングパラメータ λ_k に対して繰返し計算のなかで非零の係数を持つ予測変数の個数とし，t を更新を行う回数としたときに，提案手法のパラメータ λ_k の解に対する計算コストは $O(m_k t + np\max\{0, m_k - m_{k-1}\})$ である．

証明 Algorithm 1 に示すとおり，提案手法は予測変数の集合 $\mathcal{U}$ を設定し，式 (10) を用いて初期値を設定する．これらの処理には $O(p)$ の計算コストが必要である．そして参照ベクトルの設定を $O(m_k)$ のコストで行う．繰返し計算において必要なコストは $O(m_k t)$ となる．これは係数の更新，KKT 条件，参照ベクトルにおけるパラメータ $z[i]$ の値は式 (4) を用いることにより $O(m_t)$ で計算できるからである．なお上限値と下限値を計算するために必要な計算量は $O(t)$ となる．これは補題 3 より各繰返し計算における上限値と下限値とは $O(1)$ で計算できるからである．さらに繰返し計算において，もし新たに予測変数が非零の係数を持てば式 (4) を用いるためにその予測変数とその他全ての予測変数との間の内積を計算する必要がある．そのため，予測変数とその他全ての予測変数との内積を計算するのに必要な計算量は $O(np)$ であり，また新たに内積を計算する予測変数の数は $\max\{0, m_k - m_{k-1}\}$ であるため，式 (4) 用いるために必要な内積を計算するのに必要なコストは $O(np\max\{0, m_k - m_{k-1}\})$ となる．結果的にチューニングパラメータ λ_k に対して提案手法が必要とする計算コストは $O(m_k t + np\max\{0, m_k - m_{k-1}\})$ である． □

定理 1 では $k=0$ のとき $m_k = 0$ とする．提案手法は係数が 0 となる予測変数の更新を枝刈りできるため，Lasso を解く従来手法より高速な処理が可能となる．回帰分析の精度について提案手法は以下の性質を持つ．

定理 2 もし行列 $\mathbf{X}$ が列フルランクであれば提案手法は従来手法と同じ結果を返す．

Algorithm 1 提案手法

```
 1: for k = 1 to K do
 2:     λ := λ_k;
 3:     if k = 1 then
 4:         U := ∅;
 5:     else
 6:         U := P_{k-1};
 7:     係数の初期値を式 (10) を用いて計算;
 8:     repeat
 9:         repeat
10:             w̃_r := w̃;
11:             for each p_i ∈ U do
12:                 z̃_r[i] を w̃_r から計算;
13:             repeat
14:                 for each p_i ∈ U do
15:                     if z_[i] > λ or z̄[i] < −λ then
16:                         w̃[i] を式 (4) を用いて更新;
17:                         ‖w̃ − w̃_r‖_2 を式 (9) を用いて更新;
18:             until w̃ が収束
19:             w̃_r := w̃;
20:             for each p_i ∈ U do
21:                 z̃_r[i] を w̃_r から計算;
22:             repeat
23:                 for each p_i ∈ U do
24:                     if z̄[i] > λ or z_[i] < −λ then
25:                         w̃[i] を式 (4) を用いて更新;
26:                     else
27:                         w̃[i] = 0;
28:                     ‖w̃ − w̃_r‖_2 を式 (9) を用いて更新;
29:             until w̃ が収束
30:             for each p_i ∈ S_k do
31:                 if p_i が KKT 条件を満たさない then
32:                     p_i を U に追加;
33:         until ∀p_i ∈ S_k, p_i が KKT 条件を満たす
34:         for each p_i ∈ P do
35:             if p_i が KKT 条件を満たさない then
36:                 p_i を U に追加;
37:     until ∀p_i ∈ P, p_i が KKT 条件を満たす
```

証明 Algorithm 1 に示すとおり，提案手法はまず $\underline{z}[i] > \lambda$ または $\bar{z}[i] < -\lambda$ となるような予測変数に対して更新を行い，$\bar{z}[i] > \lambda$ または $\underline{z}[i] < -\lambda$ となる予測変数に対して更新を行う．ここで $\underline{z}[i] > \lambda$ または $\bar{z}[i] < -\lambda$ であれば必ず $\bar{z}[i] > \lambda$ または $\underline{z}[i] < -\lambda$ となる．そのため提案手法は $\bar{z}[i] \leq \lambda$ かつ $\underline{z}[i] \geq -\lambda$ となるような予測変数は更新しない．結果的に提案手法は係数が非零になるような予測変数を繰返し計算において枝刈りしない．行列 $\mathbf{X}$ が列フルランクであれば Lasso は一つの解を持つため，座標降下法に基づく手法はその一つの解に収束する．提案手法と従来手法は座標降下法に基づく手法であるため，行列 $\mathbf{X}$ が列フルランクであれば提案手法は従来手法と同様に一つの解に収束することが明らかである． □

$p > n$ である場合，行列 $\mathbf{X}$ は列フルランクにならないため，一つの解に必ずしも収束しない．しかし次の章に示すとおり，この場合において提案手法は従来手法より少ない予測変数で高い精度の回帰を行える．

4. 評価実験

提案手法の有効性を確認するために *DNA*，*Protein*，*Reuters*，*TDT2*，*Newsgroups* の 5 つのデータを用いて評価実験を行った．これらのデータにおいてデータポイントの数はそれぞれ 600，2871，8293，10212，18846 であり，特徴量の数は 180，357，18933，36771，26214 である．データの詳細は Chih-Jen Lin のウェブページ*1 と Deng Cai のウェブページ*2 に記載されている．Reuters と TDT2 と Newsgroups では $p > n$ となるため，これらのデータにおいて行列 $\mathbf{X}$ は列フルランクにならないが，その他のデータにおいて行列 $\mathbf{X}$ は列フルランクになる．実験では $\lambda_1 = \frac{1}{n}\max_i |\langle \mathbf{x}_i, \mathbf{y} \rangle|$，$\lambda_K = 0.001\lambda_1$，$K = 50$ とし，指数的にチューニングパラメータの値を減らし

∗1 http://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/multiclass.html
∗2 http://www.cad.zju.edu.cn/home/dengcai/Data/TextData.html

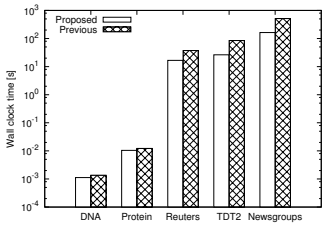

図 1: 処理時間

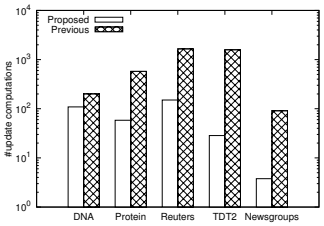

図 2: 更新回数

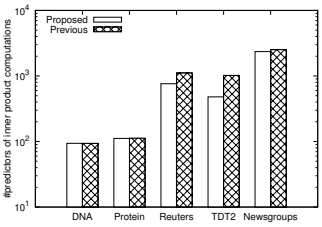

図 3: 内積計算を行った予測変数の数

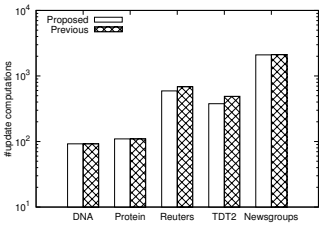

図 4: 非零となる予測変数の数

表 1: 各手法による回帰分析の結果

データセット	最小 2 乗誤差		目的関数	
	提案手法	従来手法	提案手法	従来手法
DNA	5.097×10^{-2}	5.097×10^{-2}	5.866×10^{-2}	5.866×10^{-2}
Protein	2.616×10^{-3}	2.616×10^{-3}	4.492×10^{-3}	4.492×10^{-3}
Reuters	5.003×10^{-6}	5.006×10^{-6}	7.241×10^{-6}	7.251×10^{-6}
TDT2	1.679×10^{-6}	1.691×10^{-6}	2.760×10^{-6}	2.795×10^{-6}
Newsgroups	3.463×10^{-6}	3.478×10^{-6}	4.811×10^{-6}	4.838×10^{-6}

ていった．ここで “Proposed” と “Previous” はそれぞれ提案手法と従来手法の結果を示す．実験はチューニングパラメータを変化させて行ったため，ここでは実験結果における平均値を記載する．実験は CPU が Indel Xeon 2.7 GHz の Linux サーバで行った．

4.1 計算時間

各手法における回帰分析における処理時間を評価した．処理時間を図 1 に，更新を行う計算回数を図 2 に内積計算を行った予測変数の数を図 3 に示す．図 1 から提案手法は従来手法より高速なことがわかる．これは従来手法が sequential strong rule で枝刈りされなかった全ての予測変数に対して更新を行うのに対して，提案手法は上限値と下限値を求めることで更新を行う予測変数の数を効果的に削減するためである．その結果，図 2 に見られるように提案手法は従来手法より更新における計算回数を減らすことができた．さらに図 3 に示すように提案手法は $p>n$ であるようなデータ（Reuters，TDT2，Newsgroups）に対して効果的に内積計算の回数を減らすことができた．これは後に示すように提案手法はより少ない数の予測変数で回帰分析を行えるからである．提案手法は従来手法より少ない更新計算と内積計算で回帰分析を行うため，高速な処理が可能である．

4.2 回帰分析の精度

この章では応答変数が与えられた時に，各手法における回帰分析の精度について調査した．回帰分析の結果を評価するために交差検証を用いた．表 1 に回帰分析における最小 2 乗誤差と式 (1) で与えられる目的関数の結果を示す．また図 4 に回帰分析において非零となる予測変数の数を示す．

期待したとおり行列 $\mathbf{X}$ が列フルランクとなるデータ（DNA と Protein）において，提案手法と従来手法における最小 2 乗誤差と非零となる予測変数の数は等しくなった．これは定理 2 に示したとおり，行列 $\mathbf{X}$ が列フルランクとなる場合，提案手法は従来手法と同じ回帰分析の結果となることが理論的に保証されているからである．また $p > n$ となるデータ（Reuters，TDT2，Newsgroups）に対して提案手法は従来手法よりわずかに最小 2 乗誤差が小さくなる一方，予測変数の数はより少なくなる結果ととなった．Algorithm 1 に示すとおり，提案手法はパラメータ $z[i]$ の上限値と下限値を計算し，非零となる予測変数に対して更新を行う．そのため提案手法は $|z[i]|$ が大きい値となる予測変数ほど更新する事となる．パラメータ $z[i]$ は座標降下法を行う予測変数の勾配に対応しているため [Friedman 07]，提案手法は結果的により勾配のある予測変数を優先的に更新することとなる．その結果，提案手法は上限値と下限値を用いることで式 (1) における目的関数の解を効果的に計算することができる．そのため表 1 に見られるように，提案手法における目的関数は従来手法より小さくなる結果となった．一方従来手法は予測変数を総当たり的に更新するため，最終的に零になる予測変数に対しても更新を行う事となる．表 1 及び図 1 から，提案手法は従来手法より高速により精度の高い回帰分析を行えることがわかる．

5. まとめ

Lasso は高次元データから予測変数を見つけるのに用いられる代表的な回帰分析の手法である．Lasso においてよく用いられる座標降下法は，各説明変数の重みを収束するまで総当たり的に繰返し更新することで回帰分析を行っている．しかし座標降下法による手法は高い計算量が必要になるという問題がある．本論文では繰返し計算において重みが零になる予測変数の更新を枝刈りすることで Lasso の高速化を行う手法を提案した．実験により提案手法は従来手法より効率的及び効果的に Lasso による回帰分析を行えることを確認した．

参考文献

[Friedman 07] Friedman, J., Hastie, T., Höfling, H., and Tibshirani, R.: Pathwise Coordinate Optimization, *Annals of Applied Statistics*, Vol. 1, No. 2, pp. 302–332 (2007)

[Friedman 10] Friedman, J. H., Hastie, T., and Tibshirani, R.: Regularization Paths for Generalized Linear Models via Coordinate Descent, *Journal of Statistical Software*, Vol. 33, No. 1, pp. 1–22 (2010)

[Kasiviswanathan 11] Kasiviswanathan, S. P., Melville, P., Banerjee, A., and Sindhwani, V.: Emerging Topic Detection Using Dictionary Learning, in *CIKM*, pp. 745–754 (2011)

[Liu 14] Liu, J., Zhao, Z., Wang, J., and Ye, J.: Safe Screening with Variational Inequalities and Its Application to Lasso, in *ICML*, pp. 289–297 (2014)

[Tibshirani 96] Tibshirani, R.: Regression Shrinkage and Selection Via the Lasso, *Journal of the Royal Statistical Society, Series B*, Vol. 58, pp. 267–288 (1996)

[Tibshirani 12] Tibshirani, R., Bien, J., Friedman, J., Hastie, T., Simon, N., Taylor, J., and Tibshirani, R. J.: Strong Rules for Discarding Predictors in Lasso-type Problems, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, Vol. 74, No. 2, pp. 245–266 (2012)