

交互近接勾配法とその応用 *

秋田県立大学大学院システム科学技術研究科 小野寺優希也

秋田県立大学システム科学技術学部 松下慎也

秋田県立大学システム科学技術学部 徐粒

Graduate School of Systems Science and Technology

Akita Prefectural University, Yukiya Onodera,

Faculty of Systems Science and Technology

Akita Prefectural University, Shin-ya Matsushita,

Faculty of Systems Science and Technology

Akita Prefectural University, Li Xu

1 背景

本研究では以下の最適化問題を考える：

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} \|Ax - b\|_2^2 + \lambda \|x\|_1, \quad (1)$$

ここで、 $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $\|\cdot\|_1$ は ℓ_1 ノルム, $\|\cdot\|_2$ は ℓ_2 ノルム, $\lambda > 0$ である. 式 (1) の最適化問題はラッソ回帰と呼ばれている. ラッソ回帰は曲線フィッティングや画像信号処理へ応用できる数理モデルとして知られている [3, 7].

問題 (1) の最適解を求める方法として近接勾配法が知られており, 近接勾配法で生成された点列は問題 (1) の最適解 x^* に収束する. 一方, Attouch と Peypouquet は近接勾配法の加速化法として FISTA を提案した [1]. しかし, FISTA によって生成された点列は振動しながら解に近づく問題点がある [6]. 最近, FISTA の問題点を改善するために Iutzeler と Hendrickx によって交互近接勾配法が提案された [5]. しかし, 数値例で交互近接勾配法と FISTA を比較すると, 交互近接勾配法は収束の速さは FISTA に比べて劣っている場合がある ([5, subsection 5.1], 表 4 を参照).

2 目的

本研究では交互近接勾配法に基づく新たなアルゴリズムを提案する. 具体的には交互近接勾配法と FISTA を組み合わせたアルゴリズムを提案する. 加えて, 提案手法をラッソ回帰に対して適用し, 数値実験により収束の速さを比較する.

* This work was supported by the Research Institute for Mathematical Sciences, an International Joint Usage/Research Center located in Kyoto University.

3 準備

$\gamma > 0$, $f: \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ を凸関数とする. このとき, f の近接写像 $\text{prox}_{\gamma f}: \mathbb{R}^n \rightarrow \mathbb{R}^n$ を以下のように定義する.

$$\text{prox}_{\gamma f}(x) := \arg \min_{y \in \mathbb{R}^n} \left(f(y) + \frac{1}{2\gamma} \|x - y\|_2^2 \right) \quad (2)$$

ここで, $\arg \min g(x)$ で $g(x)$ を最小とする解を表す. 本研究で用いるアルゴリズムにおいて ℓ_1 ノルムの近接写像を利用する. ℓ_1 ノルムの近接写像 $\text{prox}_{\gamma \|\cdot\|_1}$ は

$$[\text{prox}_{\gamma \|\cdot\|_1}(x)]_i = \begin{cases} x_i - \gamma & (x_i > \gamma) \\ 0 & (-\gamma \leq x_i \leq \gamma) \\ x_i + \gamma & (x_i < -\gamma) \end{cases} \quad (i = 1, 2, \dots, n) \quad (3)$$

で与えられる.

4 交互近接勾配法と提案手法

Iutzeler と Hendrickx によって提案された交互近接勾配法を紹介する [5]. 問題 (1) に対する交互近接勾配法は $k = 1, 2, \dots$ に対して

$$x_{k+1} = \text{prox}_{\gamma \lambda \|\cdot\|_1}(y_k - \gamma A^T(Ay_k - b)) \quad (4)$$

で与えられる. ただし,

$$y_k = \begin{cases} x_k & (k \text{ は偶数}) \\ x_k + \beta_k(x_k - x_{k-1}) & (k \text{ は奇数}) \end{cases} \quad (5)$$

であり, $\gamma = 1/\|A^T A\|$, $\beta_k = 0.5$ ($k = 1, 2, \dots$) である. 点列 $\{x_k\}$ を交互近接勾配法から生成された点列とし, x^* を問題 (1) の最適解としたとき, $\{x_k\}$ は x^* に収束する.

以下では交互近接勾配法と FISTA を組み合わせたアルゴリズムを紹介する. 問題 (1) に対する提案手法は Algorithm 1 で与えられる.

Algorithm 1 (提案手法). $x_0, x_1 \in \mathbb{R}^n$ を初期点, $\gamma = 1/\|A^T A\|$ とし, $\varepsilon' > 0$ を固定する. このとき, $i = 1, 2, \dots$ に対して,

$$x_{k+1} = \text{prox}_{\gamma \lambda \|\cdot\|_1}(y_k - \gamma A^T(Ay_k - b)) \quad (6)$$

である. ただし, $\|x_k - \text{prox}_{\gamma \lambda \|\cdot\|_1}(x_k - \gamma A^T(Ax_k - b))\|_2 > \varepsilon'$ のとき, 点列 $\{y_k\}$ は

$$y_k = x_k + \alpha_k(x_k - x_{k-1}) \quad (7)$$

である. ここで, 点列 $\{\alpha_k\}$ は,

$$\alpha_k = \frac{t_{k-1} - 1}{t_k}, \quad t_k = \frac{k+3}{3}, \quad t_0 = 1 \quad (8)$$

である. また, k_0 が $\|x_{k_0} - \text{prox}_{\gamma \lambda \|\cdot\|_1}(x_{k_0} - \gamma A^T(Ax_{k_0} - b))\|_2 \leq \varepsilon'$ を満たす最小の値のとき, $k \geq k_0$ に対して, 点列 $\{y_k\}$ は

$$y_k = \begin{cases} x_k & (k \text{ は偶数}) \\ x_k + \beta_k(x_k - x_{k-1}) & (k \text{ は奇数}) \end{cases} \quad (9)$$

である.

Algorithm 1 によって生成された点列 $\{x_k\}$ に関して以下の性質が成り立つ.

Theorem 1. $\{x_k\}$ を Algorithm 1 によって生成された点列とする. このとき, $\{x_k\}$ は問題 (1) の最適解に収束する.

Proof. まず, $\|x_k - \text{prox}_{\gamma\lambda\|\cdot\|_1}(x_k - \gamma A^T(Ax_k - b))\|_2$ が 0 に収束することを示す. [2, Theorem 26.1(iv)(d)] より, $\text{prox}_{\gamma\lambda\|\cdot\|_1}(x_k - \gamma A^T(Ax_k - b))$ は非拡大写像になるから,

$$\begin{aligned} & \|x_k - \text{prox}_{\gamma\lambda\|\cdot\|_1}(x_k - \gamma A^T(Ax_k - b))\|_2 \\ &= \|\text{prox}_{\gamma\lambda\|\cdot\|_1}(y_{k-1} - \gamma A^T(Ay_{k-1} - b)) - \text{prox}_{\gamma\lambda\|\cdot\|_1}(x_k - \gamma A^T(Ax_k - b))\|_2 \end{aligned} \quad (10)$$

$$\leq \|y_{k-1} - x_k\|_2 \quad (11)$$

$$= \|x_{k-1} - x_k + \alpha_k(x_{k-1} - x_{k-2})\|_2 \quad (12)$$

$$\leq \|x_{k-1} - x_k\|_2 + \alpha_k\|x_{k-1} - x_{k-2}\|_2 \quad (13)$$

ここで, [1, Theorem 1] より, $\|x_{k+1} - x_k\|_2 \rightarrow 0$ ($k \rightarrow +\infty$) であるから, $\|x_k - \text{prox}_{\gamma\lambda\|\cdot\|_1}(x_k - \gamma A^T(Ax_k - b))\|_2 \rightarrow 0$ ($k \rightarrow +\infty$) を得る. したがって, 任意の ε' に対して, ある $n_0 \in \mathbb{N}$ が存在して, $n \geq n_0$ となる $n \in \mathbb{N}$ に対して, $\|x_k - \text{prox}_{\gamma\lambda\|\cdot\|_1}(x_k - \gamma A^T(Ax_k - b))\|_2 < \varepsilon'$ が成り立つ. この x_{n_0} に対して [4, Lemma 3.3] を用いると点列 $\{x_k\}$ は問題 (1) の最適解に収束する. $\square$

5 数値実験

ラッソ回帰に対して近接勾配法 (PG), FISTA, 交互近接勾配法 (APG), 提案手法 (Proposed) を適用し結果を比較した. 本研究で用いた計算環境を表 1 に示す.

今回考えるラッソ回帰の問題設定は以下の通りである. 問題 (1) において, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ とする. ここで, A の各成分は標準正規分布に従っており, b は $b = A\hat{x} + e$ である. ただし, $\hat{x} \in \mathbb{R}^n$ は標準正規分布から得られる 10% スパースベクトルであり, $e \in \mathbb{R}^m$ の各成分は平均 0, 分散 0.001^2 の正規分布に従っている. A と b のサイズに関しては $(m, n) = (130, 80), (650, 400), (1300, 800)$ としている. 本実験では, 正則化パラメータは $\lambda = 0.1$ としている. また, 提案手法で用いる ε' は $\varepsilon' = 10^{-3}$ としている.

次に終了条件の説明をする. 問題 (1) の双対問題は [8, 式 (58)]

$$\begin{aligned} & \max_{u \in \mathbb{R}^m} \quad d_{\text{ls}}(u) := -\frac{1}{2}\|u\|_2 - \langle b, u \rangle \\ & \text{subject to} \quad \|A^T u\|_\infty \leq \lambda \end{aligned} \quad (14)$$

である. また, $h(v) = \|v - b\|_2^2/2$ としたとき, 点列 $\{u_k\}$ を

$$u_k := \min \left\{ 1, \frac{\lambda}{\|A^T \nabla h(Ax_k)\|_\infty} \right\} \nabla h(Ax_k) \quad (15)$$

とする. $f(x) = \|Ax - b\|_2/2$, $g(x) = \lambda\|x\|_1$ としたとき, 本実験の終了条件を

$$\frac{|f(x_k) + g(x_k) - d_{\text{ls}}(u_k)|}{\max\{f(x_k) + g(x_k), 1\}} \leq 10^{-8} \quad (16)$$

とする [8, section 4].

A , b を上記のように生成し, 初期点を原点とする. このとき, 終了条件 (16) を満たすまでにかかった繰り返し回数と実行時間を求めた. 各アルゴリズムに対して実験を 100 回行ない, 繰り返し回数と実行時間の平均を取ったものを比較した. A と b のサイズが $(m, n) = (130, 80), (650, 400), (1300, 800)$ の時の結果をそれぞれ表 2, 表 3, 表 4 に示す. 表 2, 表 3 より, 先行研究のアルゴリズムで一番繰り返し回数が少ないものは交互

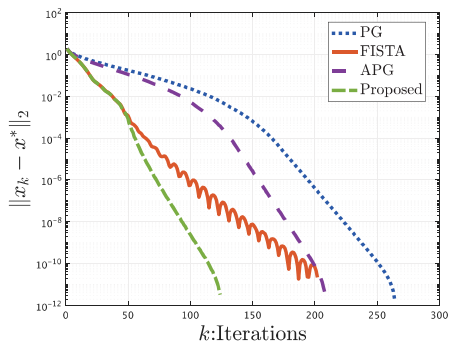


図 1 各繰り返し回数における点列 $\{x_k\}$ と近似解 x^* の差 $\|x_k - x^*\|_2$ の関係 ($m = 130, n = 80$)

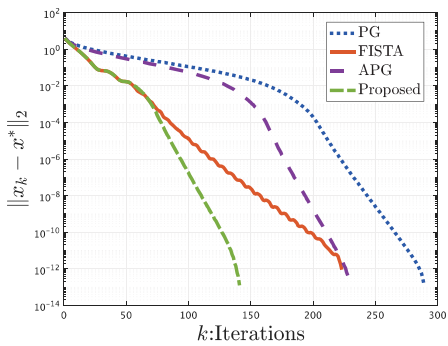


図 2 各繰り返し回数における点列 $\{x_k\}$ と近似解 x^* の差 $\|x_k - x^*\|_2$ の関係 ($m = 650, n = 400$)

近接勾配法となっている。交互近接勾配法と提案手法を比較した結果、繰り返し回数をそれぞれ 30.9 %, 36.4 % 改善することができた。表 4 より先行研究で一番繰り返し回数が少ないアルゴリズムは FISTA となっている。FISTA と提案手法を比較した結果、繰り返し回数を 34.9 % 改善することができた。

また、同様の実験を 10 回行ない、各繰り返し回数における点列 $\{x_k\}$ と近似解 x^* の差 $\|x_k - x^*\|_2$ の平均を取ったものを図示した。A と b のサイズが $(m, n) = (130, 80), (650, 400), (1300, 800)$ の時の結果をそれぞれ図 1, 図 2, 図 3 に示す。

表 1 数値実験で用いた計算環境

OS	macOS Big Sur
メモリ	8 GB 1600 MHz DDR3
CPU	1.8 GHz Intel Core i5
使用ソフト	MATLAB R2021a

表 2 各アルゴリズムに対する繰り返し回数 (Iterations) と実行時間 (CPU time) の平均の比較 ($m = 130, n = 80$)

	Iteration	CPU time[s]
PG	181.89	0.02090
FISTA	157.32	0.01782
APG	143.32	0.01609
Proposed	99.05	0.01375

表 3 各アルゴリズムに対する繰り返し回数 (Iterations) と実行時間 (CPU time) の平均の比較 ($m = 650, n = 400$)

	Iteration	CPU time[s]
PG	269.14	1.31309
FISTA	215.19	1.04852
APG	212.87	1.04364
Proposed	135.41	0.86179

表 4 各アルゴリズムに対する繰り返し回数 (Iterations) と実行時間 (CPU time) の平均の比較 ($m = 1300, n = 800$)

	Iteration	CPU time[s]
PG	309.90	10.97978
FISTA	241.51	8.53192
APG	245.41	9.11189
Proposed	157.16	7.02321

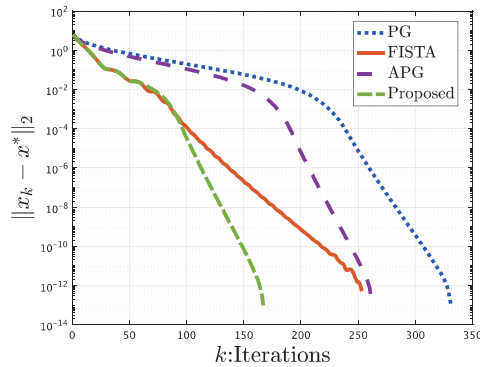


図3 各繰り返し回数における点列 $\{x_k\}$ と近似解 x^* の差 $\|x_k - x^*\|_2$ の関係 ($m = 1300, n = 800$)

6 結論

本研究ではラッソ回帰における新たなアルゴリズムを提案した。提案手法をラッソ回帰に適用した結果、先行研究のアルゴリズムと比較して繰り返し回数を 30.9 % から 36.4 % 改善することができた。

今後の課題は提案手法で用いているアルゴリズムの切り替えの評価式の ε' の決定方法を検討することである。

参考文献

- [1] H. ATTOUCH AND J. PEYPOUQUET, *The rate of convergence of nesterov's accelerated forward-backward method is actually faster than $1/k^2$* , SIAM J. Optim., 26 (2016), pp. 1824–1834.
- [2] H. H. BAUSCHKE AND P. L. COMBETTES, *Convex analysis and monotone operator theory in Hilbert spaces*, Springer, 2017.
- [3] A. CHAMBOLLE AND T. POCK, *A first-order primal-dual algorithm for convex problems with applications to imaging*, J. Math. Imaging Vision, 40 (2011), pp. 120–145.
- [4] F. IUTZELER AND J. M. HENDRICKX, *A generic online acceleration scheme for optimization algorithms via relaxation and inertia*, Optim. Methods Software, 34 (2019), pp. 383–405.
- [5] F. IUTZELER AND J. MALICK, *On the proximal gradient algorithm with alternated inertia*, J. Optim. Theory Appl., 176 (2018), pp. 688–710.
- [6] Z. MU AND Y. PENG, *A note on the inertial proximal point method*, Stat. Optim. Inf. Comput., 3 (2015), pp. 241–248.
- [7] M. NAGAHARA, *Sparsity Methods for Systems and Control*, Now Publishers, 2020.
- [8] B. WEN AND X. XUE, *On the convergence of the iterates of proximal gradient algorithm with extrapolation for convex nonsmooth minimization problems*, J. Global Optim., 75 (2019), pp. 767–787.