

交互近接勾配法とラッソ回帰への応用

Alternated proximal gradient method and its application to Lasso

○小野寺優希也*, 松下慎也*, 徐粒*

○ Yukiya Onodera*, Shin-ya Matsushita*, Li Xu*

*秋田県立大学

*Akita Prefectural University

キーワード： 交互近接勾配法 (alternated proximal gradient method), 近接勾配法 (proximal gradient method), ラッソ回帰 (Lasso)

連絡先： 〒015-0055 秋田県由利本荘市土谷字海老ノ口 84-4 秋田県立大学大学院システム科学技術研究科
電子情報システム学専攻 制御工学研究室 小野寺優希也, Tel.: (0184)27-2511,
E-mail: m20b007@akita-pu.ac.jp

1. はじめに

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. ラッソ回帰は画像信号処理や曲線フィッティングへ応用できる数理モデルとして知られている^{1, 2, 3}).

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 $\{x_k\}$ と問題 (1) の最適解 x^* には以下の関係が成り立つことが知られている.

$$\|x_{k+1} - x^*\|_2 \leq \|x_k - x^*\|_2 \quad (k = 1, 2, \dots) \quad (2)$$

式 (2) より近接勾配法は x^* に単調に収束する. 一方, Beck と Teboulle は近接勾配法の加速化法として FISTA を提案した⁴). しかし, FISTA によって生成された点列は式 (2) を満たさないという問題点がある. 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). 本研究では交互近接勾配法の有用性を検証する. 特に, この解法をラッソ回帰に応用し, その収束性を検証する.

Iutzeler と Hendrickx の数値実験では, アルゴリズムの有用性を関数値の差によって評価している^{5, 6}). この評価方法では, 関数値の誤差が小さくなっていることが確認できるもののアルゴリズムによって生成された点列 $\{x_k\}$ が最適解 x^* に収束しているかどうかを確認できない. したがって, 本研究では, アルゴリズムの評価を

生成された点列と最適解との差に関して行ない、アルゴリズムの有用性を検証する。

2. 準備

この節では次節で紹介するアルゴリズムを説明する上で必要な数学的な概念を定義する。

本論文で使用する数学記号を Table 1 に示す。また、これ以降では x^* を問題 (1) の最適解とする。

Table 1 本論文で使用する記号

記号	説明
\mathbb{R}	実数全体
\mathbb{R}^n	n 次元 Euclid 空間
$\mathbb{R}^{m \times n}$	m 行 n 列の実行列
A^T	行列 A の転置行列
$\ x\ _1$	ℓ_1 ノルム, $\ x\ _1 := \sum_{i=1}^n x_i $
$\ x\ _2$	ℓ_2 ノルム, $\ x\ _2 := \sqrt{\sum_{i=1}^n x_i^2}$
$\arg \min f(x)$	$f(x)$ が最小となる x の集合

Definition 1 (凸関数). $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ とする。関数 f が凸関数であるとは、任意の $x, y \in \mathbb{R}^n$ と $\alpha \in (0, 1)$ に対して

$$f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y) \quad (3)$$

を満たすときをいう。

Definition 2 (近接写像). $\gamma > 0$, $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ を凸関数とする。このとき、 f の近接写像 $\text{prox}_{\gamma f}$ を

$$\text{prox}_{\gamma f}(x) := \arg \min_{y \in \mathbb{R}^n} \left(f(y) + \frac{1}{2\gamma} \|x - y\|_2^2 \right) \quad (4)$$

で定義する。

次節で紹介するアルゴリズムは ℓ_1 ノルムの近接写像を計算する必要がある。 ℓ_1 ノルムの近接写像は以下のように計算できる。

Example 1. ℓ_1 ノルムの近接写像は

$i = 1, 2, \dots, n$ に対して

$$[\text{prox}_{\gamma \|\cdot\|_1}(x)]_i = \text{sgn}(x_i) \max\{|x_i| - \gamma, 0\} \quad (5)$$

で与えられる。ここで、 sgn は符号を表す。

3. アルゴリズム

この節では問題 (1) の最適解を求めるためのアルゴリズムを紹介する。

問題 (1) の最適解を求めるアルゴリズムとして、近接勾配法が知られている。近接勾配法のアルゴリズムは Algorithm 1 で与えられる。

Algorithm 1 近接勾配法

Input: $x_0 \in \mathbb{R}^n$, $\gamma \in (0, 2/\|A^T A\|)$, $\lambda > 0$

- 1: **while** 停止条件を満たすまで **do**
- 2: $x_{k+1} = \text{prox}_{\gamma \lambda \|\cdot\|_1}(x_k - \gamma A^T(Ax_k - b))$
- 3: **end while**

Output: x_k

近接勾配法の収束定理は以下のようになる⁷⁾。

Theorem 1. Algorithm 1 によって生成された点列 $\{x_k\}$ は問題 (1) の最適解に収束する。

Beck と Teboulle によって提案された FISTA のアルゴリズムは Algorithm 2 で与えられる⁴⁾。

Algorithm 2 FISTA⁴⁾

Input: $x_0, x_1 \in \mathbb{R}^n$, $\gamma \in (0, 1/\|A^T A\|)$,

$t_1 = 1$, $\lambda > 0$

- 1: **while** 停止条件を満たすまで **do**
- 2: $t_{k+1} = (1 + \sqrt{1 + 4t_k^2})/2$
- 3: $\alpha_k = (t_k - 1)/t_{k+1}$
- 4: $y_k = x_k + \alpha_k(x_k - x_{k-1})$
- 5: $x_{k+1} = \text{prox}_{\gamma \lambda \|\cdot\|_1}(y_k - \gamma A^T(Ay_k - b))$
- 6: **end while**

Output: x_k

ラッソ回帰において近接勾配法と FISTA を適用し、それぞれのアルゴリズムにおいて生成され

た点列 $\{x_k\}$ と x^* の差 $\|x_k - x^*\|_2$ を比較した結果を Fig. 1 に示す. なお, ラッソ回帰の問題設定の詳細は数値実験の節を参照すること. Fig. 1 より近接勾配法は式 (2) を満たしていることがわかる. また, FISTA は式 (2) を満たしてはいないが, 近接勾配法よりも速く x^* に収束していることがわかる. そこで, 式 (2) を満たしながら, 加速化を図るアルゴリズムである交互近接勾配法を Algorithm 3 に示す⁶⁾.

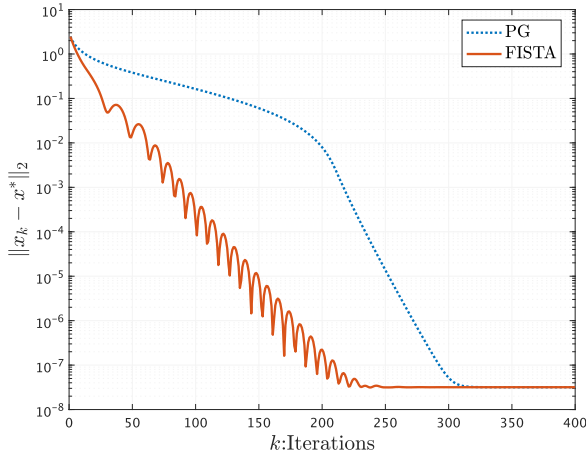


Fig. 1 ラッソ回帰に対して近接勾配法 (PG) と FISTA を適用して点列 $\{x_k\}$ と最適解 x^* の差 $\|x_k - x^*\|_2$ を比較した結果

Algorithm 3 交互近接勾配法

Input: $x_0, x_1 \in \mathbb{R}^n$, $\gamma \in (0, 2/\|A^T A\|)$,
 $\lambda > 0$, $\alpha_k \in [0, +\infty)$, ただし, $\alpha_k \leq$
 $\min\{1, 1/(\gamma\|A^T A\|)\} - 1/2$

- 1: **while** 停止条件を満たすまで **do**
- 2: **if** k is even **then**
- 3: $y_k = x_k$
- 4: **else**
- 5: $y_k = x_k + \alpha_k(x_k - x_{k-1})$
- 6: **end if**
- 7: $x_{k+1} = \text{prox}_{\gamma\lambda\|\cdot\|_1}(y_k - \gamma A^T(Ay_k - b))$
- 8: **end while**

Output: x_k

交互近接勾配法の収束定理は以下のようにな

る⁶⁾.

Theorem 2. $\{x_k\}$ を Algorithm 3 で生成された点列と, x^* を問題 (1) の最適解とする. このとき, 以下が成り立つ.

- (1) $\{x_k\}$ は x^* に収束する.
- (2) このとき, 以下が成り立つ. ただし, k は偶数である.

$$\|x_{k+2} - x^*\|_2 \leq \|x_k - x^*\|_2. \quad (6)$$

4. 数値実験

この節では, ラッソ回帰に対して近接勾配法, FISTA, 交互近接勾配法を適用し結果を比較する. 数値実験で用いた計算環境を Table 2 に示す.

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

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, $A \in \mathbb{R}^{130 \times 80}$, $b \in \mathbb{R}^{130}$ 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 $b = Ax^* + e$ である. ただし, $x^* \in \mathbb{R}^{80}$ は標準正規分布から得られる 10% スパースベクトル, $e \in \mathbb{R}^{130}$ の各成分は平均 0, 分散が 0.001 の正規分布に従っている. 本実験では, 初期値は原点としており, 正則化パラメータは $\lambda = 0.1$, ステップサイズは $\gamma = 1/(2\|A^T A\|)$ としている. また, 交互近接勾配法で使用する点列 $\{\alpha_k\}$ は $\alpha_k = 0.5$ としている.

上記の設定におけるラッソ回帰に対して近接勾配法, FISTA, 交互近接勾配法を適用し, それぞれのアルゴリズムによって生成された点列 $\{x_k\}$ と x^* の差 $\|x_k - x^*\|_2$ の比較を行なった.

このときの結果を Fig. 2 に示す。ただし、交互近接勾配法に関しては、式 (6) が成り立つことを確認するため、偶数番目からなる点列 $\{x_{2k}\}$ も図示している。また、最適解 x^* は CVX* で求めたものを使用している。加えて、Fig. 2 において近接勾配法, FISTA, 交互近接勾配法をそれぞれ, PG, FISTA, APG と表現しており、交互近接勾配法の偶数番目からなる点列を APG(even) としている。

Fig. 2 より交互近接勾配法の方が近接勾配法よりも速く解に収束していることがわかる。また、FISTA は式 (2) を満たしていないが、交互近接勾配法の偶数番目からなる点列は式 (6) が成り立っていることがわかる。

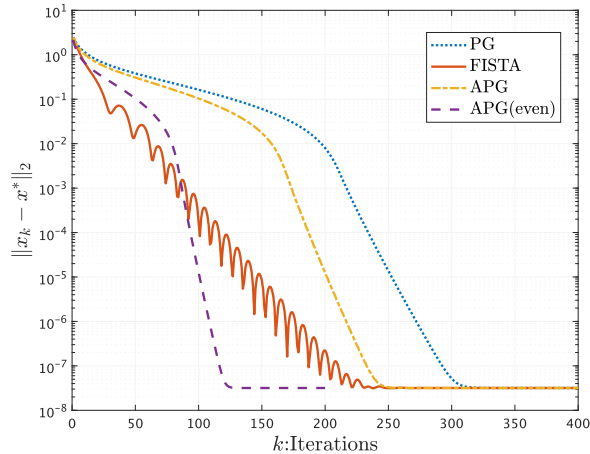


Fig. 2 ラッソ回帰に対して近接勾配法 (PG), FISTA, 交互近接勾配法 (APG) を適用して点列 $\{x_k\}$ と最適解 x^* の差 $\|x_k - x^*\|_2$ を比較した結果

5. 結論

数値実験では点列 $\{x_k\}$ と最適解 x^* の差 $\|x_k - x^*\|_2$ を比較することで、交互近接勾配法が式 (2) を満たしながら、近接勾配法よりも早く最適解に収束していることを確認することができた。

今後の課題は 2 点ある。1 点目は、クラスタリングや画像信号処理などのラッソ回帰以外の最適化問題で交互近接勾配法の有用性を検証することである。2 点目は、交互近接勾配法で生成された点列 $\{x_k\}$ に関する収束のレートの理論的な保証を得ることである。

参考文献

- 1) A. Chambolle and T. Pock: A first-order primal-dual algorithm for convex problems with applications to imaging, *J. Math. Imaging Vision*, **40**-1, 120/145 (2011)
- 2) M. Nagahara: Sparsity methods for systems and control, 31-53, Now Publishers (2020)
- 3) 鈴木讓: スパース推定 100 問 with Python, 2-22, 共立出版 (2021)
- 4) A. Beck and M. Teboulle: A fast iterative shrinkage-thresholding algorithm for linear inverse problems, *SIAM J. Imag. Sci.*, **2**-1, 183/202 (2009)
- 5) F. Iutzeler and J. M. Hendrickx: A generic online acceleration scheme for optimization algorithms via relaxation and inertia, *Optim. Methods Software*, **34**-2, 383/405 (2019)
- 6) F. Iutzeler and J. M. Hendrickx: On the proximal gradient algorithm with alternated inertia, *J. Optim. Theory Appl.*, **176**-3, 688/710 (2018)
- 7) H. H. Bauschke and P. L. Combettes: Convex analysis and monotone operator theory in Hilbert spaces, 465-480, Springer (2017)

* 凸最適化のライブラリ (<http://cvxr.com/>)