パラメータ調整不要な非凸加速勾配法

丸茂　直貴，武田　朗子

非凸最適化問題に対して，勾配と Hesse 行列の Lipschitz 連続性の仮定のもとで，$O(\varepsilon^{-7/4})$ もしくは $\tilde{O}(\varepsilon^{-7/4})$ 回の勾配計算で ε-停留点を求める一次法がいくつも提案されている．しかしこれら既存の一次法は Lipschitz 定数が既知であることを前提としており，実用性に課題がある．本研究では，既存手法と同じく $O(\varepsilon^{-7/4})$ の計算量を保証しつつも，Lipschitz 定数を入力する必要がない一次法を提案する．提案手法は Lipschitz 定数を自動的に推定するため，入力パラメータの調整をせずとも，理論上も実用上も良好な性能を示す．

キーワード：非凸最適化，一次法，Nesterov の加速勾配法，計算量解析

1. はじめに

以下の一般的な無制約最適化問題を考える．

$$\min_{x\in\mathbb{R}^d} \ f(x) \tag{1}$$

ここで，$f\colon \mathbb{R}^d \to \mathbb{R}$ は下に有界で，凸とは限らない十分滑らかな関数とする．f の非凸性によりこの問題の最適解を求めることは困難である．そのため，多くの場合，目的関数 f の勾配の Euclid ノルムが許容誤差 $\varepsilon > 0$ 以下であるような点，すなわち $\|\nabla f(x)\| \le \varepsilon$ なる点 $x \in \mathbb{R}^d$ を一つ見つけることを目標とする．このような点は **ε-停留点**と呼ばれる．本稿でも ε-停留点を求めるアルゴリズムについて議論する．

最適化問題 (1) を解くためのアルゴリズムはさまざま考えられるが，ここでは目的関数 f の情報を**オラクル**と呼ばれるブラックボックスを通してのみ取得し，その情報を用いて解を更新する反復法を考える．そのようなアルゴリズムの代表例として，最急降下法や Newton 法が挙げられる．最急降下法は，点 $x \in \mathbb{R}^d$ を与えると $f(x)$ と $\nabla f(x)$ の値を計算して返してくれるようなオラクルを用いるアルゴリズムだといえる．このような関数値・勾配オラクルを用いるアルゴリズムは，一階までの微分情報を用いるということから**一次法**と呼ばれる．Newton 法は関数値と勾配に加えて Hesse 行列も返すオラクルを用いるアルゴリズムであり，そのような手法は二次法と呼ばれる．二次法は目的関数についてより豊富な情報を活用できる分，一次法より少ない反復で収束することが期待される．一方で，問題の次元 d が大きい（$d = 10^4$ から $d = 10^5$ 程度以上）場合には，$d \times d$ 行列である Hesse 行列の計算コストが高くなるため，一次法の方が有利である．本稿では大規模な問題に対し有利な一次法に焦点を当てる．

最適化問題に適用するアルゴリズムを選択するにあたり，アルゴリズム同士の性能を理論的に比較できると便利である．アルゴリズムの理論的な性能指標としては，最悪時オラクル計算量が広く用いられている．本稿では単に**計算量**と呼ぶ．これは，

- 問題のクラス（すなわち関数 f に対する仮定）
- アルゴリズム（特に使用オラクル）
- 目標とする点（本稿では ε-停留点）

の三つから定まる値で，「問題のクラス」に含まれる問題例のうち当該アルゴリズムにとって最も都合の悪い問題例において，「目標とする点」を求めるために必要なオラクルの呼び出し回数を指す．アルゴリズムの反復回数ではなくオラクル呼び出し回数に着目するのは，一反復あたりのオラクル呼び出し回数が異なるアルゴリズムたちを公平に比較するためである．この計算量は，アルゴリズムにおいてオラクル呼び出し部分の計算コストが支配的な場合に有用な性能指標となる．

一次法の計算量保証の例を挙げる．f の勾配が Lipschitz 連続である，すなわち条件

$$\|\nabla f(x) - \nabla f(y)\| \le L\|x - y\|, \quad \forall x, y \in \mathbb{R}^d \tag{2}$$

を満たす問題クラスに対する最急降下法の計算量は $O(\varepsilon^{-2})$ である．つまり，目的関数 f が仮定 (2) を満たすならば，それがどんなに都合の悪い関数であっても，最急降下法により $O(\varepsilon^{-2})$ 回の関数値・勾配計算で ε-停留点を発見できる．これは，一次法の計算量解析の中では最も基本的な結果である．さらに 2020 年になって，どんな一次法を持ち出しても同じ仮定のもとではこの計算量を改善できないことが示された [1].

まるも　なおき，たけだ　あきこ
東京大学大学院情報理工学系研究科
〒 113–8656 東京都文京区本郷 7–3–1
marumo@mist.i.u-tokyo.ac.jp
takeda@mist.i.u-tokyo.ac.jp

 Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

f が非凸な場合は最も素朴な一次法といえる最急降下法が最適であるという事実は，f が凸な場合にはより優れた計算量を達成する一次法（Nesterov の加速勾配法 [2] など）が存在するのとは対照的であり，興味深い．

この最急降下法の「最適性」の研究と並行して，関数 f に対する仮定を強めることでより良い計算量を目指す研究も進んでいる．Carmon et al. [3] は，f の勾配の Lipschitz 連続性 (2) に加えて Hesse 行列の Lipschitz 連続性

$$\left\|\nabla^2 f(x) - \nabla^2 f(y)\right\| \leq M\|x-y\|, \quad \forall x, y \in \mathbb{R}^d \tag{3}$$

を仮定し，計算量 $\tilde{O}(\varepsilon^{-7/4})$[1] を達成する一次法を 2017 年に提案した．その後，同じ仮定のもとで同じ計算量をもつ一次法やその発展版がいくつか提案されている [4–6]．2022 年には Li and Lin [7] が計算量 $O(\varepsilon^{-7/4})$ を達成する一次法を提案し，それまでの研究 [3–6] では計算量に含まれていた log 因子を取り除くことに成功した．2024 年現在でもこの計算量上界は既存最良のものである．なお，文献 [8] では仮定 (2)，(3) のもとでの一次法の計算量下界として $\Omega(\varepsilon^{-12/7})$ が示されている．計算量の上界と下界の間には未だギャップがあり，$O(\varepsilon^{-7/4})$ より良い計算量を達成する一次法が存在するかは未解明である．

上述の一次法 [3–7] は理論的には優れたアルゴリズムだが，実用性には課題が残る．最も深刻なのは，これらのアルゴリズムは仮定 (2)，(3) に登場する Lipschitz 定数 L，M の値を既知としてアルゴリズム中で用いているということである．たとえば，文献 [6, 7] のアルゴリズムは Nesterov の加速勾配法に基づく手法だが，ステップサイズは L を，加速係数は L と M を用いて定めている．しかし，現実の最適化問題においては L や M の値は未知であることが多く，その場合には適切なパラメータの組を見つけるまで何度もアルゴリズムを実行しながらパラメータ調整をする羽目になってしまう．

さらに，既存の一次法 [3–7] は許容誤差 ε も入力として必要とし，その値を用いてアルゴリズム中のパラメータを設定する．これもアルゴリズムの実用性を損なう原因となる．というのも，ε の値をどうすればアルゴリズムが実際に高性能を発揮するかは事前に予測することが難しいためである．ε を停止条件に使うのみならば，アルゴリズムの実行前に ε を決める必要はないので問題にはならないことを注意しておく．

[1] $\tilde{O}(\cdot)$ は $\log(\varepsilon^{-1})$ の多項式因子を無視する記法である．

表 1　計算量が $O(\varepsilon^{-7/4})$ もしくは $\tilde{O}(\varepsilon^{-7/4})$ である一次法の比較

文献	決定的	$O(\varepsilon^{-7/4})$	知識不要		
			L	M	ε
[4–6]					
[3]	✓				
[7]	✓	✓			
本研究	✓	✓	✓	✓	✓

これらの課題を解決するため，本研究では新たな一次法を提案する．提案アルゴリズムは文献 [6, 7] のアルゴリズムと同様に Nesterov の加速勾配法に基づく手法であり，以下の特長をもつ．

- 仮定 (2)，(3) のもとで計算量 $O(\varepsilon^{-7/4})$ を達成する．
- Lipschitz 定数 L，M の値を自動的に推定する．それらの初期推定値を入力として与える必要があるが，それが真の Lipschitz 定数から大きく外れていても，理論上も実用上も問題ない．
- 許容誤差 ε を入力として必要としない．

表 1 は，既存手法と提案手法を比較したものである．図 1 は，Rosenbrock 関数 [9] の最小化問題

$$\min_{(x,y)\in\mathbb{R}^2} (x-1)^2 + 100(y-x^2)^2$$

を四つの手法で解いた結果を示している．図 1 が示すとおり，提案アルゴリズムは入力として与える L，M の推定値を変化させても安定した性能を発揮する．最急降下法 (GD) も L を自動推定するため安定しているが，提案アルゴリズムに比べ収束は遅い．一方で文献 [6] (JNJ2018)，[7] (LL2022) のアルゴリズムは入力された推定値を真値として扱うため，入力によっては収束が遅くなる場合やそもそも ε-停留点を見つけられない場合もある．

2. Lipschitz 定数の推定

2.1 L の推定

L と M のうち L の推定は比較的易しく，古くから知られている Armijo のバックトラッキング [11] に似た方法で実現可能である．この方法は以下の補題に基づく．

補題 1（文献 [12](Lemma 1.2.3)）．仮定 (2) が成り立つとき，任意の $x, y \in \mathbb{R}^d$ に対し，

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

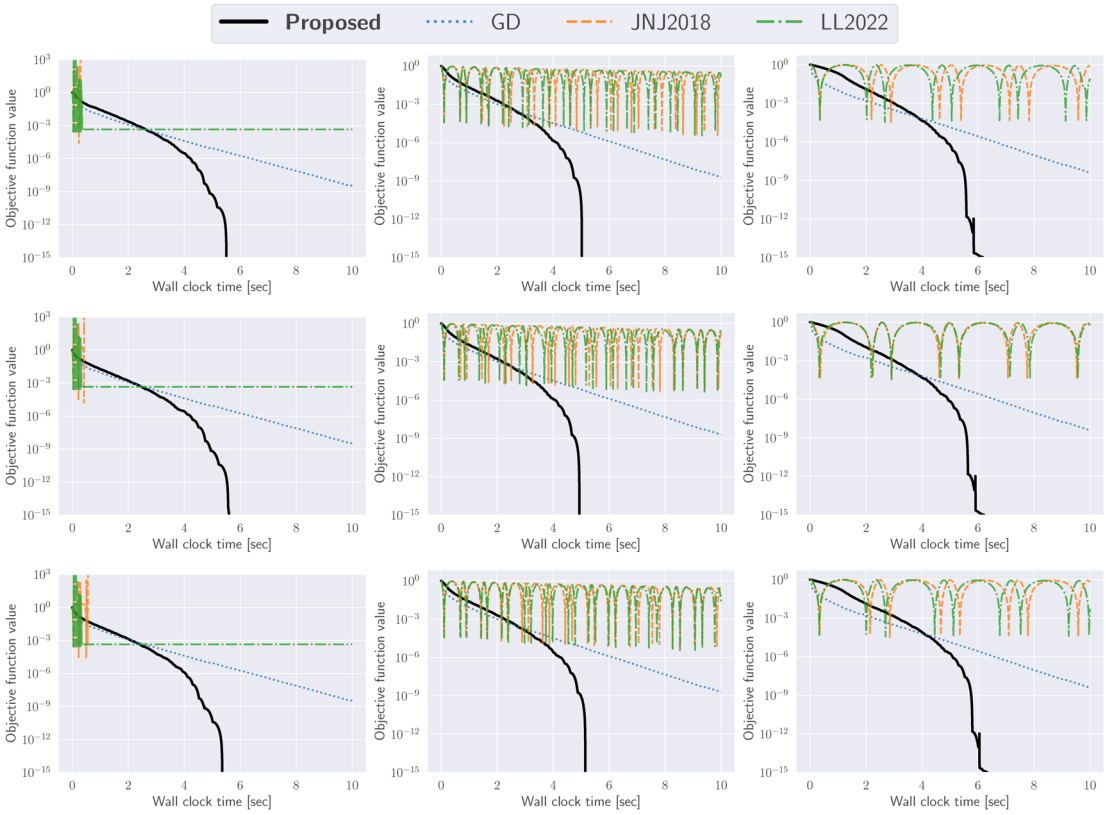


図 1　Rosenbrock 関数 [9] の最小化問題を 4 手法で解いた結果（図は文献 [10] より）
Proposed ：提案アルゴリズム，GD ：Armijo 条件によりステップサイズを選択する最急降下法，JNJ2018 ：[6](Algorithm 2)，LL2022 ：[7](Algorithm 2)．4 手法とも Lipschitz 定数 L の推定値を入力する必要があり，左から $10^2, 10^3, 10^4$ と設定した．Proposed, JNJ2018, LL2022 は M の推定値も入力する必要があり，上から $10^0, 10^1, 10^2$ と設定した．

$$f(x) - f(y) \leq \langle \nabla f(y), x - y \rangle + \frac{L}{2}\|x - y\|^2$$

が成り立つ．

Lipschitz 定数 L を既知とするアルゴリズムの計算量解析では，上の不等式の x, y にアルゴリズム中で計算される具体的な点を代入したものを用いる．L が未知の場合，アルゴリズム中の L を推定値 ℓ に置き換えることになるが，その場合も推定値 ℓ とアルゴリズム中で計算される x, y に対して

$$f(x) - f(y) \leq \langle \nabla f(y), x - y \rangle + \frac{\ell}{2}\|x - y\|^2 \quad (4)$$

が成り立つことさえ確認できれば，同様の計算量解析ができる．よって，アルゴリズム中で条件 (4) が成立するまで ℓ を大きくすることを繰り返すことで，真の Lipschitz 定数 L を用いずとも所望の計算量保証が得られる．提案アルゴリズムでは式 (4) とは異なる条件を用いるが，本質的なアイデアは同じである．

2.2　M の推定

M の推定はより非自明であり，ここが本研究で最も興味深い部分である．もしかすると，L と同様の方法で M を推定できるのではないかと思われるかもしれない．L の推定は計算量解析で用いる不等式（補題 1）に注目すると上手くいったため，ここでも M を既知とする既存の一次法の計算量解析 [3–7] で用いられている二つの不等式

$$\begin{aligned} &\|\nabla f(x) - \nabla f(y) - \nabla^2 f(y)(x - y)\| \\ &\quad \leq \frac{M}{2}\|x - y\|^2, \qquad (5) \\ &f(x) - f(y) \leq \langle \nabla f(y), x - y \rangle \\ &\quad + \frac{1}{2}\langle \nabla^2 f(y)(x - y), x - y \rangle + \frac{M}{6}\|x - y\|^3 \quad (6) \end{aligned}$$

に注目してみよう．確かにこれらの不等式において M をその推定値 m で置き換えた不等式の成立がアルゴリ

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

ズム中で確認できれば，その不等式を満たすまで m を大きくする手法がとれ，万事解決である．しかし，不等式 (5)，(6) は Hesse 行列 $\nabla^2 f(y)$ を含むため，関数値・勾配オラクルのみしか使えない一次法ではその成立を確認することができない．このように，既存の解析を基にしている限りは，L の推定と同じ手法を M の推定に用いることはできない．

そこで本研究では M の推定のために，不等式 (5)，(6) のかわりに Hesse 行列を含まない以下の二つの不等式を用いる．

補題 2. 仮定 (3) が成り立つとき，任意の $x, y \in \mathbb{R}^d$ に対し，

$$f(x) - f(y) \leq \frac{1}{2}\langle \nabla f(x) + \nabla f(y), x - y\rangle + \frac{M}{12}\|x - y\|^3$$

が成り立つ（図 2）．

補題 3. 仮定 (3) が成り立つとする．$z_1, \ldots, z_n \in \mathbb{R}^d$ と $\sum_{i=1}^n \lambda_i = 1$ なる $\lambda_1, \ldots, \lambda_n \geq 0$ を任意にとり，$\bar{z} := \sum_{i=1}^n \lambda_i z_i$ とおく．このとき，

$$\left\| \nabla f(\bar{z}) - \sum_{i=1}^n \lambda_i \nabla f(z_i) \right\| \leq \frac{M}{2} \sum_{i=1}^n \lambda_i \|z_i - \bar{z}\|^2$$

が成り立つ．

われわれは，Lipschitz 定数 M が登場する不等式としてはこれらのみを用いる新たな計算量解析手法を開発した．それに基づき M を推定するアルゴリズムを提案する．

補題 2 の不等式は，数値解析の分野では台形則の誤差評価式として知られている不等式 [13](式 (5.1.4)) の多次元版と見なせる．この補題は補題 1 と似た形をしているが，不等式の最後の項が $O(\|x - y\|^2)$ ではなく $O(\|x - y\|^3)$ になっていることが重要である．これは大雑把に言って，x と y が十分近いときには，$\langle \nabla f(y), x - y\rangle$ よりも $\frac{1}{2}\langle \nabla f(x) + \nabla f(y), x - y\rangle$ の方が遥かに良く $f(x) - f(y)$ を近似することを表している．この近似精度の向上は仮定 (3) からもたらされたものであり，これは $O(\varepsilon^{-2})$ より良い計算量を得るための鍵の一つである．

補題 3 の不等式は，凸関数 $\phi\colon \mathbb{R}^d \to \mathbb{R}$ に対する Jensen の不等式

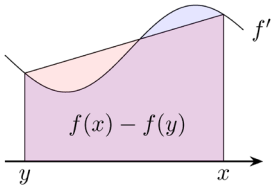


図 2 補題 2 のイメージ
曲線で囲まれた部分の面積は $f(x) - f(y)$ であり，台形の面積は $\frac{1}{2}\langle \nabla f(x) + \nabla f(y), x - y\rangle$ である．それらの差は $\frac{M}{12}\|x - y\|^3$ で押さえられる．

$$\phi(\bar{z}) - \sum_{i=1}^n \lambda_i \phi(z_i) \leq 0$$

によく似た形をしている．Jensen の不等式は，重み付き平均をとる操作と関数 ϕ をかませる操作の順番を入れ替えたときの値の変化量を評価したものだが，補題 3 は，重み付き平均をとる操作と勾配をとる操作の順番を入れ替えたときのベクトルの変化量を評価している．補題 3 は，平均解 $\bar{z}$ における勾配 $\nabla f(\bar{z})$ を，勾配の平均 $\sum_{i=1}^n \lambda_i \nabla f(z_i)$ で近似したときの誤差を与える不等式とも見なせる．凸最適化においては，平均解 $\bar{z}$ での目的関数値 $f(\bar{z})$ が小さいことを Jensen の不等式を使って示す証明技法がしばしば用いられるが（文献 [14] など），本研究では平均解での勾配のノルム $\|\nabla f(\bar{z})\|$ が小さいことを補題 3 から示す．なお，補題 3 の不等式に類似した不等式が文献 [15](Corollary 2) で示されている．

3. 提案アルゴリズム

提案する一次法をアルゴリズム 1 に示す．本アルゴリズムは, Nesterov の加速勾配法 [2] を基礎としている．

解の更新（行 4，5） ∇f の Lipschitz 定数 L の推定値 ℓ が与えられたとき，加速勾配法では点列 (x_k)，(y_k) を以下の更新式で定める．

$$x_k = y_{k-1} - \tfrac{1}{\ell}\nabla f(y_{k-1}),$$
$$y_k = x_k + \theta_k(x_k - x_{k-1}).$$

ここで，$y_0 = x_0$ であり，$0 \leq \theta_k \leq 1$ は加速係数である．アルゴリズムの性能を高めるためには θ_k の定め方が重要だが，本研究では

$$\theta_k = \frac{k}{k+1} \tag{7}$$

を提案する．これは，非凸関数に対する既存の加速勾配法で用いられている $\theta_k = 1 - (M\varepsilon)^{1/4}/(4\sqrt{L})$ [6]

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

や $\theta_k = 1 - 2(M\varepsilon)^{1/4}/\sqrt{L}$ [7] とは大きく異なる．重要なのは，式 (7) は Lipschitz 定数 L，M や許容誤差 ε を含まないという点である．

平均解（行 6）　行 6 では解 $y_0, \ldots, y_{k-1}$ の重み付き平均を計算している．計算量解析において，平均解での勾配 $\nabla f(\bar{y}_k)$ のノルムが小さいことを，補題 3 を用いて示す．

再始動機構（行 9，11）　行 9 もしくは行 11 の条件を満たしたとき，$k = 0$，$y_0 = x_0$ と設定し直す．この挙動を再始動と呼ぶ．行 9 の再始動機構は，推定値 ℓ の更新のためのものであり，Armijo のバックトラッキングと同様の働きをしている．詳細は割愛するが，$\ell \geq L$ ならば行 9 の条件は決して成立しないことが証明でき，アルゴリズム中で常に $\ell \leq \max\{\ell_{\mathrm{init}}, 2L\}$ が成立することがわかる．このように，推定値 ℓ が上界をもつことは計算量解析の際に重要になる．

行 9 の再始動機構は，主に補題 3 の不等式の右辺を小さく保つためのものである．この再始動条件の中でのみ M の推定値 m_k が使われている．この m_k は，補題 2，3 に基づいて設計された更新式

$$m_k = \max\Bigg\{m_{k-1}, 4\frac{(k+1)^2\|\nabla f(\bar{y}_k)\| - 4\ell\|x_k - x_{k-1}\|}{(k-1)(k+5)^2 S_k}, 12\frac{f(y_k) - f(x_k) - \frac{1}{2}\langle \nabla f(y_k) + \nabla f(x_k), y_k - x_k\rangle}{\|y_k - x_k\|^3}, \frac{\|(k+1)\nabla f(y_k) + k\nabla f(x_{k-1}) - (2k+1)\nabla f(x_k)\|}{k\|x_k - x_{k-1}\|^2}\Bigg\} \tag{8}$$

により計算する．この式は複雑に見えるが，これは単に計算量解析で必要になる四つの不等式

$$\begin{aligned}
&m_{k-1} \leq m,\\
&(k+1)^2\|\nabla f(\bar{y}_k)\| - 4\ell\|x_k - x_{k-1}\| \leq \frac{m}{4}(k-1)(k+5)^2 S_k,\\
&f(y_k) - f(x_k) \leq \frac{1}{2}\langle \nabla f(y_k) + \nabla f(x_k), y_k - x_k\rangle + \frac{m}{12}\|y_k - x_k\|^3,\\
&\|(k+1)\nabla f(y_k) + k\nabla f(x_{k-1}) - (2k+1)\nabla f(x_k)\| \leq mk\|x_k - x_{k-1}\|^2
\end{aligned}$$

をすべて満たす最小の m を計算するだけである．上の不等式に登場する x_k や y_k は計算済みであるため，そのような m はバックトラッキングなどを用いることなく式 (8) のように明示的に書ける．また，補題 2，3 より，このように計算された m_k は必ず $m_k \leq \max\{m_0, M\}$ を満たすことが示せる．

アルゴリズム 1　再始動機構付き加速勾配法

Input: $x_{\mathrm{init}} \in \mathbb{R}^d$; $\ell_{\mathrm{init}}, m_0 > 0$
1: $(x_0, y_0) \leftarrow (x_{\mathrm{init}}, x_{\mathrm{init}})$, $\ell \leftarrow \ell_{\mathrm{init}}$, $k \leftarrow 0$
2: **repeat**
3: $\quad k \leftarrow k + 1$
4: $\quad x_k \leftarrow y_{k-1} - \frac{1}{\ell}\nabla f(y_{k-1})$
5: $\quad y_k \leftarrow x_k + \frac{k}{k+1}(x_k - x_{k-1})$
6: $\quad \bar{y}_k \leftarrow \frac{2}{k(k+1)}\sum_{i=0}^{k-1}(i+1)y_i$
7: $\quad S_k \leftarrow \sum_{i=1}^{k}\|x_i - x_{i-1}\|^2$
8: $\quad M$ の推定値 m_k を式 (8) により計算
9: $\quad$ **if** $f(x_k) > f(x_0) - \frac{\ell S_k}{2(k+1)}$ **:**
10: $\quad\quad (x_0, y_0) \leftarrow (x_{k-1}, x_{k-1})$, $\ell \leftarrow 2\ell$, $k \leftarrow 0$
11: $\quad$ **else if** $(k+1)^5 m_k^2 S_k > \ell^2$ **:**
12: $\quad\quad (x_0, y_0) \leftarrow (x_k, x_k)$, $k \leftarrow 0$
13: **until** convergence
14: **return** $\bar{y}_k$

4. 計算量保証

前節で示した推定値 ℓ，m_k の上界をそれぞれ $\bar{L} := \max\{\ell_{\mathrm{init}}, 2L\}$，$\bar{M} := \max\{m_0, M\}$ とおく．提案アルゴリズムの反復回数について，以下の結果が示せる．

定理 1.　仮定 (2), (3) が成り立つとし，$\Delta := f(x_{\mathrm{init}}) - \inf_{x\in\mathbb{R}^d} f(x)$ とおく．このとき，アルゴリズム 1 において初めて $\|\nabla f(\bar{y}_k)\| \leq \varepsilon$ が満たされるまでに必要な合計反復回数は

$$\frac{70\Delta\bar{L}^{1/2}\bar{M}^{1/4}}{\varepsilon^{7/4}} + O(\varepsilon^{-1/4})$$

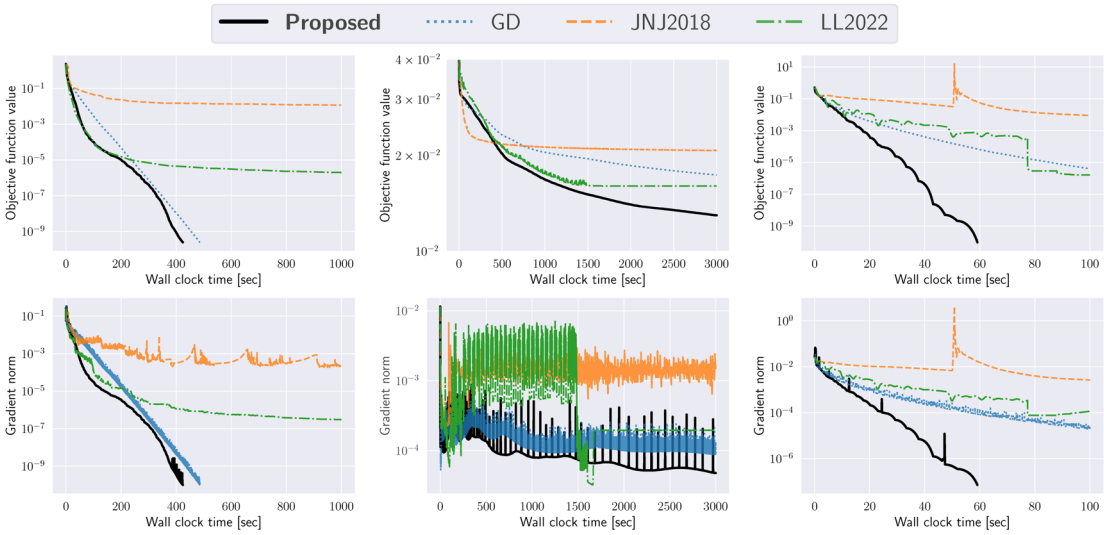
以下である．

この定理は，提案アルゴリズムが入力パラメータに依らず計算量 $O(\varepsilon^{-7/4})$ を達成することを示している．Lipschitz 定数についての知識をもとにパラメータを設定しないと計算量が保証されない既存の一次法とは大きく異なる．

5. 数値実験

機械学習の分野で登場する三つの問題例を用いて，提案アルゴリズムの性能を検証した．最初の二つの問

 Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.



(a) 分類　(b) 自己符号化器　(c) 行列補完

図 3　数値実験結果
比較手法は図 1 と同じ．提案アルゴリズムでは $\ell_{\mathrm{init}} = 10^{-3}$，$m_0 = 10^{-16}$ とした．JNJ2018 と LL2022 では，文献 [7] に倣って $M = 1$，$\varepsilon = 10^{-16}$ とし，L は $\{10^{-4}, 10^{-3}, \ldots, 10^{4}\}$ の中から最も性能が良くなるものを選んだ．

題例は，分類と自己符号化器に関するもので，それぞれ

$$\min_{w \in \mathbb{R}^d} \frac{1}{N} \sum_{i=1}^{N} \ell_{\mathrm{CE}}(y_i, \phi_1(x_i; w)), \tag{9}$$

$$\min_{w \in \mathbb{R}^d} \frac{1}{2MN} \sum_{i=1}^{N} \|x_i - \phi_2(x_i; w)\|^2 \tag{10}$$

と定式化される．ここで，$x_1, \ldots, x_N \in \mathbb{R}^M$ はデータベクトルであり，$y_1, \ldots, y_N \in \{0,1\}^K$ はラベルを表すベクトルである．ℓ_{CE} は交差エントロピー損失関数，$\phi_1(\cdot; w) : \mathbb{R}^M \to \mathbb{R}^K$ と $\phi_2(\cdot; w) : \mathbb{R}^M \to \mathbb{R}^M$ はパラメータ $w \in \mathbb{R}^d$ をもつニューラルネットワークである．これらの問題では，MNIST データセット [16] の手書き数字画像から $N = 10{,}000$ 個のデータをランダムに選んで用いた．このデータセットでは $K = 10$，$M = 784$ である．ニューラルネットワークのパラメータ数は，問題 (9) では $d = 25{,}818$，問題 (10) では $d = 52{,}064$ とした．

もう一つの問題例は，行列補完問題であり，

$$\min_{\substack{U \in \mathbb{R}^{p \times r} \\ V \in \mathbb{R}^{q \times r}}} \sum_{(i,j,s) \in \Omega} \left((UV^\top)_{ij} - s \right)^2 + \left\| U^\top U - V^\top V \right\|_{\mathrm{F}}^2$$

と定式化される．Ω はデータ行列の N 個の観測成分からなり，要素 $(i,j,s) \in \Omega$ は (i,j) 成分の値が s であることを表す．この部分的に観測された行列を，階数 r の行列 $UV^\top$ で近似する．Frobenius ノルム $\|\cdot\|_{\mathrm{F}}$ を含む正則化項は，U と V のバランスをとるために文献 [17] で提案されたものである．この問題では，MovieLens-100K データセット [18] を用いた．このデータセットでは $p = 943$，$q = 1{,}682$，$N = 100{,}000$ である．階数は $r = 200$ とした．この問題のパラメータ数は $(p+q)r = 525{,}000$ である．

数値実験の結果を図 3 に示す．いずれの問題においても，提案アルゴリズムはパラメータ調整をすることなく高速に動作することが確認できた．

6. おわりに

本稿は文献 [10] に基づく．証明などの詳細はそちらを参照されたい．また後続研究として，パラメータ調整不要でより強い理論保証をもつ一次法が文献 [19] で提案されている．

謝辞　本研究は，JSPS 科研費 (JP19H04069)，JST ERATO (JPMJER1903) の助成を受けたものです．本稿の執筆の機会をくださった関谷和之先生（成蹊大学）と鮏川矩義先生（法政大学）に感謝いたします．

参考文献

[1] Y. Carmon, J. C. Duchi, O. Hinder and A. Sidford, “Lower bounds for finding stationary points I,” *Mathematical Programming*, **184**, pp. 71–120, 2020.
[2] Y. Nesterov, “A method for solving a convex pro-

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

gramming problem with convergence rate $O(1/k^2)$," *Soviet Mathematics Doklady*, **269**, pp. 372–376, 1983.
[3] Y. Carmon, J. C. Duchi, O. Hinder and A. Sidford, "Convex until proven guilty: Dimension-free acceleration of gradient descent on non-convex functions," In *Proceedings of the 34th International Conference on Machine Learning*, pp. 654–663, 2017.
[4] Y. Xu, R. Jin and T. Yang, "NEON+: Accelerated gradient methods for extracting negative curvature for non-convex optimization," *arXiv*, arXiv:1712.01033, 2017.
[5] Z. Allen-Zhu and Y. Li, "NEON2: Finding local minima via first-order oracles," In *Advances in Neural Information Processing Systems 31*, pp. 3720–3730, 2018.
[6] C. Jin, P. Netrapalli and M. I. Jordan, "Accelerated gradient descent escapes saddle points faster than gradient descent," In *Proceedings of the 31st Conference On Learning Theory*, pp. 1042–1085, 2018.
[7] H. Li and Z. Lin, "Restarted nonconvex accelerated gradient descent: No more polylogarithmic factor in the $O(\varepsilon^{-7/4})$ complexity," In *Proceedings of the 39th International Conference on Machine Learning*, pp. 12901–12916, 2022.
[8] Y. Carmon, J. C. Duchi, O. Hinder and A. Sidford, "Lower bounds for finding stationary points II: First-order methods," *Mathematical Programming*, **185**, pp. 315–355, 2021.
[9] H. H. Rosenbrock, "An automatic method for finding the greatest or least value of a function," *The Computer Journal*, **3**, pp. 175–184, 1960.
[10] N. Marumo and A. Takeda, "Parameter-free accelerated gradient descent for nonconvex minimization," *SIAM Journal on Optimization*, **34**, pp. 2093–2120, 2024.
[11] L. Armijo, "Minimization of functions having Lipschitz continuous first partial derivatives," *Pacific Journal of Mathematics*, **16**(1), pp. 1–3, 1966.
[12] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, Springer, 2004.
[13] K. E. Atkinson, *An Introduction to Numerical Analysis*, John Wiley & Sons, 2nd edition, 1989.
[14] A. Chambolle and T. Pock, "On the ergodic convergence rates of a first-order primal–dual algorithm," *Mathematical Programming*, **159**, pp. 253–287, 2016.
[15] S. Zlobec, "Jensen's inequality for nonconvex functions," *Mathematical Communications*, **9**, pp. 119–124, 2004.
[16] L. Deng, "The MNIST database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Processing Magazine*, **29**, pp. 141–142, 2012.
[17] S. Tu, R. Boczar, M. Simchowitz, M. Soltanolkotabi and B. Recht, "Low-rank solutions of linear matrix equations via Procrustes flow," In *Proceedings of the 33rd International Conference on Machine Learning*, pp. 964–973, 2016.
[18] F. M. Harper and J. A. Konstan, "The MovieLens datasets: History and context," *ACM Transactions on Interactive Intelligent Systems*, **5**, Article No. 19, 2015.
[19] N. Marumo and A. Takeda, "Universal heavy-ball method for nonconvex optimization under Hölder continuous Hessians," *Mathematical Programming*, DOI: 10.1007/s10107-024-02100-4, 2024.

 Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.