日本オペレーションズ・リサーチ学会
2021年 春季研究発表会

DC 最適化問題に対する Bregman 距離を用いた近接アルゴリズム

05001365 総合研究大学院大学 *高橋翔大 TAKAHASHI Shota
01507430 University of São Paulo 福田光浩 FUKUDA Mituhiro
02303360 統計数理研究所 田中未来 TANAKA Mirai

1. はじめに

Difference of Convex (DC) 最適化問題とは, 凸関数同士の差で表せる目的関数を扱う非凸最適化問題の一種である. 非凸最適化問題では大域的最適解はおろか, 局所的最適解を求めることでさえ困難とされている. 本研究では, 凸関数 $f_1, f_2, g : \mathbb{R}^d \to (-\infty, +\infty]$ を用いて

$$(\mathcal{P}) \quad \min_{x\in\mathbb{R}^d} \quad \Psi(x) = g(x) + f_1(x) - f_2(x),$$

と表せる DC 最適化問題を扱う. ただし, f_1 は $\mathcal{C}^1$ 級, f_2 は連続だが $\mathcal{C}^1$ 級とは限らない, g は $\mathcal{C}^1$ 級とは限らず, アルゴリズム内の子問題が解きやすい関数を選ぶ.

大規模な DC 最適化問題 $(\mathcal{P})$ を解く代表的なアルゴリズムとしては, Proximal DC Algorithm (PDCA) が知られている. PDCA は f_1 が L-smooth な関数 (∇f_1 が定数 L で Lipschitz 連続) である必要がある. 本研究では, f_1 により一般化した性質を仮定し, 幅広い問題に対応できるアルゴリズムを提案する.

2. 提案アルゴリズム

2.1. 数学的準備

カーネル生成距離 [1] と Bregman 距離を導入し, カーネル生成距離を用いて, L-smooth 性を一般化した L-smad [1] を定義する.

定義 2.1. 非空な開凸集合 C に対し, 以下の条件を満たす $h : \mathbb{R}^d \to (-\infty, \infty]$ をカーネル生成距離という.

(i) h は下半連続真凸関数で $\mathrm{dom}\, h \subset \overline{C}, \mathrm{dom}\, \partial h = C$.

(ii) h は $\mathrm{int\ dom}\, h = C$ 上で $\mathcal{C}^1$ 級.

また, $\mathcal{G}(C)$ をカーネル生成距離の集合とする. カーネル生成距離 $h \in \mathcal{G}(C)$ に対し, Bregman 距離 $D_h : \mathrm{dom}\, h \times \mathrm{int\ dom}\, h \to \mathbb{R}_+$ を以下のように定義する.

$$D_h(x, y) = h(x) - h(y) - \langle \nabla h(y), x - y \rangle.$$

定義 2.2. 以下の関数の組 (f, h) を考える.

(i) $h \in \mathcal{G}(C)$.

(ii) 下半連続真関数 $f : \mathbb{R}^d \to (-\infty, +\infty]$ が, $\mathrm{dom}\, h \subset \mathrm{dom}\, f$ を満たし, $C = \mathrm{int\ dom}\, h$ 上で $\mathcal{C}^1$ 級.

組 (f, h) が C 上で L-smooth adaptable (L-smad) であるとは, C 上で $Lh - f, Lh + f$ が凸関数となる定数 $L > 0$ が存在することである.

以降, 関数の組 (f_1, h) は L-smad であると仮定する.

2.2. 提案アルゴリズム: BPDCA

DC 最適化問題 $(\mathcal{P})$ に対して, Bregman Proximal DC Algorithm (BPDCA) を提案する.

Algorithm 1 BPDCA

入力: $x^0 \in \mathbb{R}^d, \lambda \in (0, \frac{1}{L})$.
for $k = 0, 1, 2, \cdots,$ **do**
$\xi^k \in \partial f_2(x^k)$,

$$x^{k+1} = \underset{x\in\mathbb{R}^d}{\arg\min} \{ g(x) + \langle \nabla f_1(x^k) - \xi^k, x - x^k \rangle + \frac{1}{\lambda} D_h(x, x^k) \}.$$

また, $h(x) = \frac{1}{2}\|x\|^2$ のとき, $D_h(x, x^k) = \frac{1}{2}\|x - x^k\|^2$ となるので, BPDCA は PDCA に一致する. つまり, BPDCA は PDCA を一般化したアルゴリズムである. 多くの g, h の組み合わせで子問題を簡単に解くことができる.

2.3. 提案アルゴリズム: BPDCAe

アルゴリズムを改善するために, 加速パラメータを導入し, Bregman Proximal DC Algorithm with Extrapolation (BPDCAe) を提案する.

Algorithm 2 BPDCAe

入力: $x^0 = x^{-1} \in \mathbb{R}^d, \{\beta_k\}_{k=0}^{\infty}, \lambda \in (0, \frac{1}{L})$.
for $k = 0, 1, 2, \cdots,$ **do**
$\xi^k \in \partial f_2(x^k)$,
$y^k = x^k + \beta_k(x^k - x^{k-1})$,

$$x^{k+1} = \underset{y\in\mathbb{R}^d}{\arg\min} \{ g(y) + \langle \nabla f_1(y^k) - \xi^k, y - y^k \rangle + \frac{1}{\lambda} D_h(y, y^k) \}.$$

加速パラメータ β_k は, $\theta_0 = \theta_{-1} = 1$ で初期化し,

$$\beta_k = \frac{\theta_{k-1} - 1}{\theta_k} < 1, \quad \theta_{k+1} = \frac{1 + \sqrt{1 + 4\theta_k^2}}{2},$$

で与える. さらに, 以下の条件

(i) あらかじめ定めた定数 $K \in \mathbb{N}$ に対し, $k = tK, \quad \forall t \in \mathbb{N}$,

(ii) あらかじめ定めた定数 $\rho \in [0, 1)$ に対し, $D_h(x^k, y^k) > \rho D_h(x^{k-1}, x^k)$

のどちらかを満たすとき, $\theta_{k-1} = \theta_k = 1$ とする.

3. 収束定理

3.1. 収束定理: BPDCA

BPDCA の生成する点列に関して, 次の補題が成り立つ.

補題 3.1. 点列 $\{x^k\}_{k=0}^{\infty}$ を BPDCA の生成する点列とすると, 以下の不等式が成り立つ:

$$\lambda\Psi(x^{k+1}) \leq \lambda\Psi(x^k) - (1 - \lambda L) D_h(x^{k+1}, x^k).$$

特に, $0 < \lambda L < 1$ のとき, 目的関数値 Ψ は減少する.

目的関数 が Kurdyka-Łojasiewicz (KL) 関数であると仮定し, 補題 3.1 より, 次の定理を示すことができる.

定理 3.2. 目的関数 Ψ は KL 関数とする. 点列 $\{x^k\}_{k=0}^{\infty}$ を BPDCA によって生成された点列とすると, 以下の事実が成立する:

(i) $\lim_{k\to\infty} \mathrm{dist}(0, \partial\Psi(x^k)) = 0$.

(ii) $\{x^k\}_{k=0}^{\infty}$ は Ψ の停留点に収束し, $\sum_{k=1}^{\infty} \|x^k - x^{k-1}\| < \infty$ となる.

3.2. 収束定理: BPDCAe

BPDCA の場合とは異なり, BPDCAe の生成する点列が目的関数を減少させることを直接示すことができない. そこで, 適当な定数 $M > 0$ を用いて補助関数を $H_M(x, y) = \Psi(x) + M D_h(y, x)$ と定義し, 補助関数が減少することを示す.

補題 3.3. 点列 $\{x^k\}_{k=0}^{\infty}$ を BPDCAe の生成する点列とすると, 以下の不等式が成り立つ:

$$\begin{aligned}\lambda\Psi(x^{k+1}) \leq& \lambda\Psi(x^k) + D_h(x^k, y^k) \\ &- D_h(x^k, x^{k+1}) - (1 - \lambda L) D_h(x^{k+1}, y^k).\end{aligned}$$

さらに, 次の不等式が成り立つ:

$$\begin{aligned}H_M(x^{k+1}, x^k) \leq& H_M(x^k, x^{k-1}) \\ &- \left(\frac{1}{\lambda} - M\right) D_h(x^k, x^{k+1}) \\ &- \left(M - \frac{\rho}{\lambda}\right) D_h(x^{k-1}, x^k).\end{aligned}$$

特に, $\frac{\rho}{\lambda} \leq M \leq \frac{1}{\lambda}, \rho \in [0, 1)$ のとき, 補助関数値 H_M は減少する.

定理 3.2 よりも複雑にはなるが, 補助関数 H_M が KL 関数であると仮定することで, 補題 3.3 より BPDCAe の生成する点列が目的関数 Ψ の停留点への収束を示すことができる.

定理 3.4. 補助関数 H_M は KL 関数で $\frac{\rho}{\lambda} \leq M \leq \frac{1}{\lambda}, \rho \in [0, 1)$ を満たすものとし, 点列 $\{x^k\}_{k=0}^{\infty}$ を BPDCAe の生成する点列とすると, 以下の事実が成立する:

(i) $\lim_{k\to\infty} \mathrm{dist}((0, 0), \partial H_M(x^k, x^{k-1})) = 0$.

(ii) $\lim_{k\to\infty} H_M(x^k, x^{k-1}) = \zeta$, $\zeta := \lim_{k\to\infty} \Psi(x^k)$.

(iii) $H_M \equiv \zeta$ が $\Upsilon := \{(x, x) | x \in \Omega\}$ 上で成り立つ. ただし, Ω は $\{x^k\}_{k=0}^{\infty}$ の集積点の集合.

(iv) $\{x^k\}_{k=0}^{\infty}$ は Ψ の停留点に収束し, $\sum_{k=1}^{\infty} \|x^k - x^{k-1}\| < \infty$ となる.

4. 数値実験

DC 最適化問題 $(\mathcal{P})$ の応用例として, Phase Retrieval がある. 既存手法である Bregman Proximal Gradient method (BPG) [1], Bregman Proximal Gradient method with Extrapolation (BPGe) [2] と比較すると, BPDCA, BPDCAe はステップサイズを大きくとることができる. このことから, 計算機上でも, 提案手法は収束が速かった. さらに, 加速パラメータを導入した BPDCAe は最も収束が速かった. 詳細は当日示す.

5. おわりに

L-smooth 性を持たない関数を含む DC 最適化問題 $(\mathcal{P})$ に対するアルゴリズムとして, BPDCA, BPDCAe を提案した. BPDCA, BPDCAe の生成する点列が目的関数の停留点に収束することを示した. BPDCAe が効率的に DC 最適化問題の停留点を得ることができるアルゴリズムであることを明らかにした.

参考文献

[1] J. Bolte, S. Sabach, M. Teboulle, and Y. Vaisbourd: First order methods beyond convexity and lipschitz gradient continuity with applications to quadratic inverse problems, *SIAM Journal on Optimization*, 28(3):2131–2151, 2018.

[2] X. Zhang, R. Barrio, M. Angeles Martinez, H. Jiang, and L. Cheng: Bregman proximal gradient algorithm with extrapolation for a class of nonconvex nonsmooth minimization problems, *IEEE Access*, 7:126515–126529, 2019.