数理解析研究所講究録 第2108巻 2019年 187-195

Inexact proximal memoryless quasi-Newton methods for minimizing composite functions

東京理科大学　中山 舜民 (Shummin Nakayama)
Tokyo University of Science
横浜国立大学　成島 康史 (Yasushi Narushima)
Yokohama National University
東京理科大学　矢部 博 (Hiroshi Yabe)
Tokyo University of Science

1　はじめに

本稿では，以下の最適化問題に対する近接勾配法を考える．

$$\min_{x\in R^n} \quad f(x) := g(x) + h(x). \tag{1.1}$$

ここで，$g : R^n \to R$ は連続的微分可能な凸関数とし，$h : R^n \to R$ は凸関数ではあるが，必ずしも微分可能とは限らないとする．関数 g の勾配 ∇g は Lipschitz 連続であるとする．すなわち，ある正定数 L が存在して

$$\|\nabla g(u) - \nabla g(v)\| \leq L\|u - v\| \quad {}^{\forall} u, v \in R^n, \tag{1.2}$$

が成り立つ．ただし，$\|\cdot\|$ は ℓ_2 ノルムとする．問題 (1.1) は g を損失関数，h を正則化項とすれば以下の例題などの様々なモデルに帰着される．

例題 1.1 $A \in R^{m\times n}$, $b \in R^m$ とし $\lambda > 0$ とする．以下の問題を *LASSO* と呼ぶ．

$$\min_{x\in R^n} \quad \frac{1}{2}\|Ax - b\|^2 + \lambda\|x\|_1.$$

ここで，$\|\cdot\|_1$ は ℓ_1 ノルムとする．

例題 1.2 $b = (b_1, \cdots, b_m)^T \in R^m$, $w = (w_1, \cdots, w_m)^T \in R^m$ とし $\lambda > 0$ とする．以下の問題を *Logistic regression* と呼ぶ．

$$\min_{x\in R^n} \frac{1}{m}\sum_{i=1}^{m} \log\left(1 + \exp(-b_i x^T w_i)\right) + \lambda\|x\|_1. \tag{1.3}$$

上記の問題に対する数値解法として，近接勾配法 [12] が知られている．近接勾配法は (1.2) に現れる Lipschitz 定数 L を用いて，任意の初期点 $x_0 \in R^n$ から出発し，反復式

$$\begin{aligned} x_{k+1} &= \operatorname*{argmin}_{x\in R^n} \left(g(x_k) + \nabla g(x_k)^T(x - x_k) + \frac{L}{2}\|x - x_k\|^2 + h(x) \right) \\ &= \operatorname*{argmin}_{x\in R^n} \left(h(x) + \frac{L}{2}\left\| x - \left(x_k - \frac{1}{L}\nabla g(x_k) \right) \right\|^2 \right) \end{aligned} \tag{1.4}$$

により点列 $\{x_k\}$ を生成する方法である．ここで近接写像を

$$\mathrm{Prox}_h(\bar{x}) \equiv \operatorname*{argmin}_{x\in R^n}\left(h(x)+\frac{1}{2}\|x-\bar{x}\|^2\right)$$

と定義すれば，(1.4) は

$$\begin{aligned} x_{k+1} &= \operatorname*{argmin}_{x\in R^n}\left(\frac{1}{L}h(x)+\frac{1}{2}\left\|x-\left(x_k-\frac{1}{L}\nabla g(x_k)\right)\right\|^2\right)\\ &= \mathrm{Prox}_{\frac{1}{L}h}\left(x_k-\frac{1}{L}\nabla g(x_k)\right)\end{aligned}$$

と表せる．しかしながら，一般に Lipschitz 定数 L は未知のため，直線探索を用いて L を代用する方法が使われる．また，近接勾配法の拡張としてニュートン型近接勾配法 [3] が知られている．ニュートン型近接勾配法は

$$x_{k+1} = x_k + \alpha_k d_k \tag{1.5}$$

により点列を生成する反復法である．ここで，$\alpha_k > 0$ をステップ幅，$d_k \in R^n$ を探索方向とする．ニュートン型近接勾配法の探索方向は重み付き近接写像に基づいており，凸関数 h に対する正定値対称な重み行列 A が付いた近接写像は

$$\mathrm{Prox}_h^A(\bar{x}) \equiv \operatorname*{argmin}_{x\in R^n}\left(h(x)+\frac{1}{2}\|x-\bar{x}\|_A^2\right) \tag{1.6}$$

で定義される．ここで $\|x\|_A = \sqrt{x^TAx}$ とする．ニュートン型近接勾配法は，重み付き近接写像と正定値対称な行列 G_k を用いて

$$x_k^+ = \mathrm{Prox}_h^{G_k}(x_k - G_k^{-1}\nabla g(x_k)) \tag{1.7}$$

を計算し，探索方向を

$$d_k = x_k^+ - x_k \tag{1.8}$$

で与える．具体的な G_k の選び方として関数 g の x_k でのヘッセ行列 $\nabla^2 g(x_k)$ やその近似行列を用いることが一般的である．ニュートン型近接勾配法の近接写像の計算において，G_k として単位行列 I に Lipschitz 乗数 L をかけた行列，つまり $G_k = LI$ を選び，$\alpha_k = 1$ とすれば，通常の近接勾配法 (1.4) に帰着されることに注意する．

近年，最適化問題の規模が増加しており，ニュートン法のような，行列を陽に利用する方法を適用することが難しい．また，一般に，重み付き近接写像 (1.6) を厳密に計算することが困難なため，非厳密ニュートン型近接勾配法 [4, 6] が提案されている．そこで，本研究では行列を陽に使用しないメモリーレス準ニュートン法 [10] に基づいた非厳密ニュートン型近接勾配法を提案し，その収束性を議論する．最後に，簡単な数値実験の結果を紹介する．

2　提案法

本節では，2.1 節で非厳密ニュートン型近接勾配法を紹介をする．次に，2.2 節ではメモリーレス準ニュートン法に基づいた非厳密ニュートン型近接勾配法を提案し，2.3 節で収束性を議論する．

2.1 非厳密ニュートン型近接勾配法

一般に，重み付き近接写像 (1.6) の正確な計算が困難であるため，ニュートン型近接勾配法の探索方向 (1.7)–(1.8) は，反復法を用いて計算される．そのため，探索方向の計算に時間を要してしまい，全体の計算時間が増大する．このことから，非厳密に近接写像を計算する非厳密ニュートン型近接勾配法 [4, 6] が提案された．近接写像を厳密に解く (1.7) の場合，(1.6) の最適性条件より

$$0 \in \nabla g(x_k) + G_k(x_k^+ - x_k) + \partial h(x_k^+)$$

が成り立つ．ここで，$\partial h(x_k^+)$ は x_k^+ での h の劣微分を表すものとする．B_k を $\nabla^2 g(x_k)$ の近似行列，$H_k = B_k^{-1}$ として，関係式

$$x_k^+ \approx \mathrm{Prox}_h^{B_k}(x_k - H_k \nabla g(x_k)) \tag{2.1}$$

を考える．このとき，上記の関係式は (1.7) の近似であるため，

$$r_k \in \nabla g(x_k) + B_k(x_k^+ - x_k) + \partial h(x_k^+) \tag{2.2}$$

という残差 r_k が存在する．非厳密ニュートン型近接勾配法は，この残差 r_k が

$$\|r_k\|_{H_k} \leq (1 - \theta_k)\|d_k\|_{B_k}, \quad \theta_k \in [\bar{\theta}, 1] \tag{2.3}$$

を満たすとき，(1.7) の代わりに (2.1) を用いて探索方向を計算する．条件 (2.3) は $\theta_k = 1$ のとき $r_k = 0$ となり，近接写像を厳密に解いていることに注意する．

2.2 メモリーレス MBFGS 法

本節では，(2.1) における B_k の選択法について議論する．準ニュートン法に基づいたニュートン型近接勾配法においては BFGS 公式

$$B_k = B_{k-1} - \frac{B_{k-1}s_{k-1}(B_{k-1}s_{k-1})^T}{s_{k-1}^T B_{k-1} s_{k-1}} + \frac{y_{k-1}y_{k-1}^T}{s_{k-1}^T y_{k-1}}$$

を用いることが一般的である．ただし

$$s_k = x_k - x_{k-1}, \quad y_{k-1} = \nabla g(x_k) - \nabla g(x_{k-1})$$

とする．この公式はセカント条件

$$B_k s_{k-1} = y_{k-1}, \tag{2.4}$$

を満たす準ニュートン法の更新式のひとつである．その他の有名な更新式としては DFP 公式や対称ランクワン (Symmetric Rank-one; SR1) 公式などがある [8]．本研究において，一様正定値かつ有界であるような B_k を考える必要がある．そのために，通常のセカント条件 (2.4) の代わりに Li and Fukushima [5] によって提案された修正セカント条件と Cheng and Li [2] によって提案されたスペクトラルスケーリングセカント条件を考える．はじめに，Cheng and Li [2] に基づいてスケーリングパラメータ $\gamma_k > 0$ を用いて

$$\gamma_k g(x) \approx \gamma_k \left(g(x_k) + \nabla g(x_k)^T (x - x_k) + \frac{1}{2}(x - x_k)^T \nabla^2 g(x_k)(x - x_k) \right)$$

という二次近似を考える．ここで上式を x で微分し，$x = x_{k-1}$ とおくことで

$$\gamma_k \nabla^2 g(x_k) s_{k-1} \approx \gamma_k y_{k-1}$$

が得られる．次に Li and Fukushima [5] に基づいて，両辺に $\gamma_k \nu_k s_{k-1}$ を加えることで

$$\gamma_k (\nabla^2 g(x_k) + \nu_k I) s_{k-1} \approx \gamma_k \left(y_{k-1} + \nu_k s_{k-1} \right)$$

が得られる．ただし $\nu_k \geq 0$ はパラメータである．さらに B_k を $\gamma_k(\nabla^2 g(x_k) + \nu_k I)$ の近似行列とし

$$z_{k-1} = y_{k-1} + \nu_k s_{k-1}$$

とすると，新たに

$$B_k s_{k-1} = \gamma_k z_{k-1} \tag{2.5}$$

というセカント条件が得られる．本研究では γ_k として，ある正定数 $\underline{\gamma}, \overline{\gamma}$ が存在するような

$$\underline{\gamma} \leq \gamma_k \leq \overline{\gamma}, \tag{2.6}$$

を選び，ν_k として，ある正定数 $\bar{\nu} \in (0, 1]$ が存在して

$$s_{k-1}^T z_{k-1} = s_{k-1}^T (y_{k-1} + \nu_k s_{k-1}) \geq \bar{\nu} \|s_{k-1}\|^2 \tag{2.7}$$

を満たすような ν_k を選ぶ．例えば

$$\nu_k = \begin{cases} 0, & \text{if } s_{k-1}^T y_{k-1} \geq \bar{\nu} \|s_{k-1}\|^2, \\ \bar{\nu} \left(1 - \frac{s_{k-1}^T y_{k-1}}{\|s_{k-1}\|^2} \right), & \text{otherwise} \end{cases}$$

とすれば (2.7) が成り立つ．

条件 (2.5) を満たす BFGS 公式は

$$B_k = B_{k-1} - \frac{B_{k-1} s_{k-1} (B_{k-1} s_{k-1})^T}{s_{k-1}^T B_{k-1} s_{k-1}} + \gamma_k \frac{z_{k-1} z_{k-1}^T}{s_{k-1}^T z_{k-1}}$$

で与えられる．さらに，メモリーレス準ニュートン法 [10] の考え方に基づいて $B_{k-1} = I$ とすれば

$$B_k = I - \frac{s_{k-1} s_{k-1}^T}{s_{k-1}^T s_{k-1}} + \gamma_k \frac{z_{k-1} z_{k-1}^T}{s_{k-1}^T z_{k-1}} \tag{2.8}$$

が得られる．さらに，(2.8) の逆行列 H_k は

$$H_k = I + \left(\frac{1}{\gamma_k} + \frac{z_{k-1}^T z_{k-1}}{s_{k-1}^T z_{k-1}} \right) \frac{s_{k-1} s_{k-1}^T}{s_{k-1}^T z_{k-1}} - \frac{z_{k-1} s_{k-1}^T + s_{k-1} z_{k-1}^T}{s_{k-1}^T z_{k-1}} \tag{2.9}$$

で与えられる．以下の命題は行列 (2.8)–(2.9) が一様正定値かつ有界な行列であることを意味している．

命題 2.1 行列 B_k と H_k はそれぞれ (2.8) と (2.9) で与えられるとする．(1.2) と (2.6), (2.7) が成り立つとする．このとき，ある正定数 $\underline{m}$ と $\overline{m}$ が存在して

$$\underline{m}\|u\|^2 \leq \|u\|^2_{B_k} = u^T B_k u \leq \overline{m}\|u\|^2 \quad {}^\forall u \in R^n,$$

$$\frac{1}{\overline{m}}\|u\|^2 \leq \|u\|^2_{H_k} = u^T H_k u \leq \frac{1}{\underline{m}}\|u\|^2 \quad {}^\forall u \in R^n$$

が成り立つ．

次に，提案法のアルゴリズムを述べる．

アルゴリズム 1 (提案法のアルゴリズム)

Step 0: 初期点 $x_0 \in R^n$ を与え，$\delta \in (0,1)$, $\beta_k \in (0,1)$, $\bar{\theta} \in (0,1]$, $k=0$ とする．

Step 1: 終了条件を満たすならばアルゴリズムは停止して x_k を最適解とする．

Step 2: (2.8) と (2.9) により B_k と H_k を求める．

Step 3: θ_k を与え，条件 (2.2)–(2.3) を満たす x_k^+ を求め，探索方向 $d_k = x_k^+ - x_k$ を計算する．

Step 4: $\alpha \in \{1, \beta, \beta^2, \ldots\}$ の中で *Armijo* 条件

$$f(x_k + \alpha d_k) \leq f(x_k) + \delta\alpha(\nabla g(x_k)^T d_k + h(x_k^+) - h(x_k)), \tag{2.10}$$

を満たす最も大きな α の値を α_k とする．

Step 5: 点 x_{k+1} を (1.5) によって更新する．

Step 6: $k = k+1$ として *Step1* に戻る．□

実際には Step 2 において行列 B_k, H_k を計算するのではなく，$\mathrm{Prox}_h^{B_k}(x_k - H_k\nabla g(x_k))$ や $\|d_k\|_{B_k}$ などの計算はベクトルの内積のみで表現できるため，提案したアルゴリズムは，大規模な問題に適用可能であることを注意しておく．また，アルゴリズム 1 の Step 4 は，バックトラッキング法と呼ばれる直線探索である．

2.3 提案法の収束性

本節では提案法の収束性を議論する．はじめに，直線探索に関する補題を与える．

補題 2.1 問題 (1.1) について考える．アルゴリズム 1 により点列 $\{x_k\}$ が生成されるとする．このとき

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \alpha_k(\nabla g(x_k)^T d_k + h(x_k^+) - h(x_k)) + O(\alpha_k^2),$$

$$\nabla g(x_k)^T d_k + h(x_k^+) - h(x_k) \leq -\bar{\theta} d_k^T B_k d_k$$

が成立する．

命題 2.1 より B_k が正定値行列であるため，降下条件

$$\nabla g(x_k)^T d_k + h(x_k^+) - h(x_k) < 0$$

が満たされるので，上記の補題は Armijo 条件 (2.10) が適用可能であることを保証している．次の補題は，アルゴリズム 1 において，直線探索が実行可能であることを保証している．

補題 2.2 問題 (1.1) について考える．(1.2) が成り立つとし，アルゴリズム 1 により点列 $\{x_k\}$ が生成されるとする．このとき，すべての $k \geq 0$ に対して *Armijo* 条件 (2.10) を満たす α_k が存在し，さらに

$$\bar{\alpha} \equiv \beta \min \left\{ 1, \frac{2\underline{m}}{L} \bar{\theta}(1-\delta) \right\} \leq \alpha_k \leq 1$$

が成り立つ．

これらの補題から以下の結論を得る．

命題 2.2 アルゴリズム 1 により点列 $\{x_k\}$ が生成されるとする．このとき x_k が (1.1) の最適解であるための必要十分条件は $d_k = 0$ である．

定理 2.1 問題 (1.1) は少なくとも 1 つの最適解を持つとする．このとき，アルゴリズム 1 により生成される点列 $\{x_k\}$ が有界ならば，点列 $\{x_k\}$ は最適解に収束する．

次に，局所的な収束性について考える．そのために μ-強凸関数を定義する．ある正定数 μ に対して不等式

$$g(u) \geq g(v) + \nabla g(v)^T (u-v) + \frac{\mu}{2}\|u-v\|^2 \quad {}^\forall u, v \in R^n$$

が成り立つ関数を μ-強凸関数という．問題 (1.1) において，関数 g が μ-強凸関数ならば，最適解 x^* は一意に存在する．

補題 2.3 アルゴリズム 1 により点列 $\{x_k\}$ が生成されるとする．このとき

$$f(x_k^+) \leq f(x_k) + \left(\frac{L}{2\underline{m}} - \bar{\theta} \right) \|d_k\|_{B_k}^2$$

が成り立つ．さらに g が μ-強凸関数であれば

$$\|x_k^+ - x^*\|^2 \leq \frac{2}{\mu} \left((f(x_k) - f(x^*)) + \left(\frac{L}{2\underline{m}} - \bar{\theta} \right) \|d_k\|_{B_k}^2 \right)$$

を得る．

補題 2.3 を用いることにより，次の定理を得る．

定理 2.2 問題 (1.1) について考える．g が μ-強凸関数であり，点列 $\{x_k\}$ はアルゴリズム 1 により生成されるとする．このとき点列 $\{x_k\}$ は x^* に以下の意味で R-1 次収束する．

$$\|x_k - x^*\| \leq \rho^k \sqrt{\frac{2}{\mu}\left(f(x_0) - f(x^*)\right)}$$

ここで $\rho \in (0,1)$ は定数である．

3 数値実験

本節では提案法の数値実験結果を報告する．本数値実験では問題 (1.3) を解いており，実験は MATLAB_R2017b を用いて行った．問題 (1.3) の b_i, w_i は UCI Machine Learning Repository [11] から選んでおり，$\lambda = 0.001$ としている．表 1 はテスト問題の問題名，データ数，次元数を表している．いずれの問題も初期点を $x_0 = (0, \cdots, 0)^T$ としている．終了条件は $\|d_k\|_\infty < 10^{-6}$ とし，直線探索のパラメータは $\delta = 0.0001$, $\beta = 0.5$ としている．ここで，$\|\cdot\|_\infty$ は ℓ_∞ ノルムである．また，探索方向を求める際，FISTA [1] を使って (2.1) を求めている．

表 1: テスト問題

問題名	データ数 m	次元数 n
a9a	32,561	123
leukemia	38	7129

はじめに，条件 (2.3) のパラメータ θ_k を変化させたときの性能を比較した．具体的に θ_k の値を 0.1 ∼ 1.0 の範囲で 0.1 刻みで比較した．ただし，$\theta_k = 1.0$ の場合は (2.2)–(2.3) を満たす x_k^+ を求めるのが困難であるため，終了条件を条件 (2.3) の代わりに $\|r_k\|_{H_k} \leq 10^{-6}$ とした．表 2 は問題 a9a に対する実験結果を表しており，“外部反復” がアルゴリズム 1 の反復回数，“内部反復” は (2.1) を求める際の FISTA の総反復回数，“計算時間 (秒)” はアルゴリズム 1 が収束するまでに費やした時間を意味している．表 2 から，$\theta_k = 1.0$ の場合は近接写像を正確に求めるために計算時間を費やしていしまって内部反復が多くなり，結果的に一番時間がかかっている．外部反復で比較すると $\theta_k = 0.9$ が一番良い結果を示していて，$\theta_k = 0.1$ の場合が一番悪い結果を示している．一方で，$\theta_k = 0.2$ の場合が内部反復が一番少なく，計算時間が一番短くなっている．計算時間の観点から，近接写像を厳密に計算する必要はないと考えられるが，非厳密に解く分だけ反復回数が多くなってしまうので，θ_k の調整が必要である．

表 2: 提案法を用いて問題 a9a を解いた結果

θ_k	外部反復	内部反復	計算時間（秒）
0.1	**203**	37739	5.14
0.2	172	**33477**	**4.39**
0.3	171	54014	5.92
0.4	177	48190	5.51
0.5	162	43769	5.02
0.6	160	37521	4.51
0.7	151	45074	4.98
0.8	155	39570	4.62
0.9	**140**	40090	4.55
1.0	152	**99005**	**8.83**

次に，提案法と既存のニュートン型近接勾配法 (PNOPT [9]) との比較をする．この方法は Lee et al. [3] で提案されたニュートン型近接勾配法である．表 3 は BFGS 公式に基づいたニュートン型近接勾配法で問題 a9a を解いた結果である．この結果から，反復回数は既存の方法が少ないものの，計算時間は提案法の方が短いことがわかる．

表 3: PNOPT を用いて問題 a9a を解いた結果

方法	外部反復	計算時間（秒）
PNOPT(BFGS)	136	6.87

次に，問題 leukemi を提案法と PNOPT で解いたときの比較を行う．その結果は表 3 で与えられる．“L-BFGS” は記憶制限 BFGS 法 [7] に基づいた方法であり，“mem” は記憶するメモリの本数を意味している．また，“BFGS” は BFGS 公式に基づいた方法である．PNOPT (BFGS) は行列を毎回更新し，行列の演算を行なっているため，1800 秒費やしても問題を解くことができなかった．これは毎回の行列の演算に時間がかかったためである．一方で PNOPT (L-BFGS, mem=10) が一番速く問題を解いている．反復回数では PNOPT (L-BFGS, mem=50) が一番少ない．提案法は反復回数が多くなってしまってる分だけ計算時間を費やしてしまっているが，既存手法とは異なり近接写像を計算するための重み行列の構造が単純なため，その計算を効率的に行うことができれば，既存手法よりも短い計算時間で問題を解くことができると期待される．

表 4: 問題 leukemia を解いた結果

方法	外部反復	計算時間（秒）
提案法 ($\theta_k = 0.8$)	3300	188.55
PNOPT (L-BFGS, mem=10)	1910	79.30
PNOPT (L-BFGS, mem=50)	328	106.69
PNOPT (BFGS)	—	1800+

謝辞

本研究の一部は JSPS 科研費 JP17K00039，および，京都大学数理解析研究所の助成を受けて行われている．

参考文献

[1] A. Beck and M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, *SIAM Journal on Imaging Sciences*, **2** (2009), 183-202.

[2] W.Y. Cheng and D.H. Li, Spectral scaling BFGS method, *Journal of Optimization Theory and Applications*, **146** (2010), 305–319.

[3] J.D. Lee, Y. Sun and, M.A. Saunders, Proximal Newton-type methods for minimizing composite functions, *SIAM Journal on Optimization*, **24** (2014), 1420–1443.

[4] J. Li, M.S. Andersen and L. Vandenberghe, Inexact proximal Newton methods for self-concordant functions, *Mathematical Methods of Operations Research*, **85** (2017), 19–41.

[5] D.H. Li and M. Fukushima, A modified BFGS method and its global convergence in nonconvex minimization, *Journal of Computational and Applied Mathematics*, **129** (2001), 15–35.

[6] X. Liu, C.J. Hsieh, J.D. Lee and Y. Sun, An inexact subsampled proximal Newton-type method for large-scale machine learning, *arXiv preprint arXiv:1708.08552*, 2017.

[7] D.C. Liu and J. Nocedal, On the limited memory method for large-scale optimization, *Mathematical Programming*, **45** (1989), 503–528.

[8] J. Nocedal and S.J. Wright, "Numerical optimization", 2^{nd} edition, Springer, 2006.

[9] PNOPT website, https://web.stanford.edu/group/SOL/software/pnopt/, (最終アクセス日：2018 年 11 月 30 日).

[10] D.F. Shanno, Conjugate gradient methods with inexact searches, *Mathematics of Operations Research*, **3** (1978), 244–256.

[11] UCI Machine Learning Repository, https://archive.ics.uci.edu/ml/datasets.html, (最終アクセス日：2018 年 11 月 30 日).

[12] 金森敬文, 鈴木大慈, 竹内一郎, 佐藤一誠, 機械学習のための連続最適化, 講談社, 2016.