DCアプローチに基づくスパース最適化

後藤　順哉，武田　朗子

本稿では非凸最適化問題に対する一次法の適用例として，ℓ_0 制約をもった最適化問題を取り上げる．特に筆者らの研究をベースに，ℓ_0 制約を二つの凸関数の差 (DC) を用いた制約式に置き換える方法，および，制約式を置き換えた問題に対するいくつかの一次法アルゴリズム，そして階数制約への拡張について紹介する．

キーワード：スパース最適化，ℓ_0 制約，DC 最適化，近接 DCA，最大 K ノルム，階数制約，Ky Fan K ノルム

1. ℓ_0 制約付き最適化

ベクトルや行列を決定変数にもつような最適化問題において，最適解ベクトルの 0（ゼロ）である要素の数を多くしたり，行列において階数を小さくしたりといったように，一部の情報だけを用いた解表現を指向する数理最適化は**スパース最適化** (sparse optimization) と総称される．スパース最適化に分類される問題が画像・信号処理，機械学習，バイオインフォマティクス，金融工学などさまざまな文脈で現れることから近年注目されている[1]．本稿では，ℓ_0 制約付き最適化問題という基本的なスパース最適化問題を中心に，DC (Difference of two Convex functions) と呼ばれる特殊な構造をもった問題に対する一次法に基づくアプローチを紹介する．

n 次元ベクトル $\boldsymbol{w}=(w_1,\dots,w_n)^\top\in\mathbb{R}^n$ の非ゼロ要素数を $\|\boldsymbol{w}\|_0$，すなわち，

$$\|\boldsymbol{w}\|_0 \;:=\; \#\{i\in\{1,\dots,n\} : w_i\neq 0\}$$

と書く．$K<n$ なる自然数 K を用いて表される条件

$$\|\boldsymbol{w}\|_0\le K \tag{1}$$

は ℓ_0 **制約**と呼ばれ，これを満たすベクトル $\boldsymbol{w}$ は少なくとも $n-K$ 個の要素がゼロとなる．ここでゼロ要素の多さがスパース性であり，K が小さいほど高いスパース性を表す[2]．適当な目的関数 $f:\mathbb{R}^n\to\mathbb{R}$ と集合 $S\subset\mathbb{R}^n$ に対して定義される最適化問題：

$$\begin{array}{ll}\underset{\boldsymbol{w}}{\text{minimize}} & f(\boldsymbol{w})\\ \text{subject to} & \|\boldsymbol{w}\|_0\le K,\ \boldsymbol{w}\in S,\end{array} \tag{2}$$

のように，(1) を制約条件にもつような最適化問題 (2) は，ℓ_0 制約付き最適化問題などと呼ばれるスパース最適化問題の典型例である．具体的な例として，n 個の特徴量からなる入力 $\tilde{\boldsymbol{x}}\in\mathbb{R}^n$ と出力 $\tilde{y}\in\mathbb{R}$ の m 組の観測標本 $(\boldsymbol{x}_1,y_1),\dots,(\boldsymbol{x}_m,y_m)\in\mathbb{R}^n\times\mathbb{R}$ から線形モデル $\tilde{y}=\boldsymbol{w}^\top\tilde{\boldsymbol{x}}=\sum_{j=1}^n w_j\tilde{x}_j$ を最小二乗法で推定する際，n 個の特徴量のうち高々 K 個までしか使わないという条件（**基数制約**とも呼ばれる）を付すような問題は

$$\begin{array}{ll}\underset{\boldsymbol{w},J}{\text{minimize}} & \sum_{i=1}^m(y_i-\sum_{j\in J}w_jx_{ij})^2\\ \text{subject to} & J\subset\{1,\dots,n\},\ \#J\le K,\end{array}$$

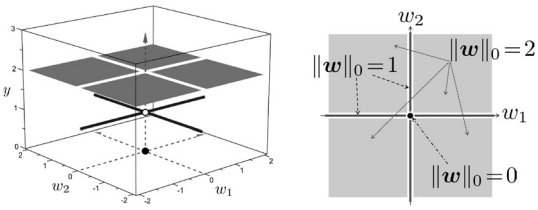
のように，組合せ的な要素をもった最適化問題として定式化される．もし $w_j=0$ であれば，入力の第 j 特徴量のデータ x_{ij}，$i=1,\dots,m$, は目的関数の減少に寄与しないため，第 j 特徴量 $\tilde{x}_j$ が線形モデルに含まれないのと同じである．このことに注意すれば，この問題は $f(\boldsymbol{w})=\sum_{i=1}^m(y_i-\boldsymbol{w}^\top\boldsymbol{x}_i)^2$, $S=\mathbb{R}^n$ とすることで (2) の ℓ_0 制約付き問題に帰着できる．

関数 $\nu(\boldsymbol{w})=\|\boldsymbol{w}\|_0$ は ℓ_0 **ノルム**と呼び習わされるが，図 1 に示すように，原点や軸のところで不連続な関数であり，したがって凸でもない[3]．実際 $\|\boldsymbol{w}\|_0\le 1$ を満たす $\boldsymbol{w}=(w_1,w_2)^\top$ は w_1 軸と w_2 軸上の点からなる十字形となり，非凸集合をなす（図 1(ii)）．この

ごとう　じゅんや
中央大学理工学部
〒 112–8551 東京都文京区春日 1–13–27
jgoto@indsys.chuo-u.ac.jp
たけだ　あきこ
東京大学情報理工学研究科
〒 113–8656 東京都文京区本郷 7–3–1
理化学研究所革新知能統合研究 (AIP) センター
〒 103–0027 東京都中央区日本橋 1–4–1
takeda@mist.i.u-tokyo.ac.jp

[1] 信号処理におけるスパース最適化については本特集小野氏の記事 [1] を参照していただきたい．
[2] (1) を満たすとき $\boldsymbol{w}$ は K **疎** (K-sparse) であると言う．
[3] 斉次性を満たさないためいわゆる「ノルム」ではない．このため ℓ_0 擬ノルムと呼ぶ流儀もある [1].

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



(i) $y = \|(w_1, w_2)\|_0$　　(ii) 等高線

図 1　$\mathbb{R}^2$ 上の ℓ_0 ノルムの概形 ($n = 2$) と等高線

ため先の基数制約付き最小二乗推定の例のように，仮に f が凸関数，S が凸集合であっても (2) は非凸最適化問題になってしまい，大域的最適性の保証を得るのは一般に容易でない．

非凸最適化問題 (2) の大域的最適解を得ることを諦め，そこそこの性能の解を効率的に求めるので十分な場合にはさまざまなアプローチがありうる．それらの中で近年人気を集めているのが，非凸性の根源と言える ℓ_0 ノルムに代えて，ℓ_1 ノルムを制限する方法，すなわち適当な定数 $\kappa > 0$ に対して，

$$\begin{aligned} &\underset{\boldsymbol{w}\in S}{\text{minimize}} && f(\boldsymbol{w}) \\ &\text{subject to} && \|\boldsymbol{w}\|_1 \le \kappa, \end{aligned} \tag{3}$$

あるいは，定数 $\eta > 0$ に対して

$$\underset{\boldsymbol{w}\in S}{\text{minimize}} \quad f(\boldsymbol{w}) + \eta\|\boldsymbol{w}\|_1 \tag{4}$$

の解を代理として用いるというものである．ここで $\boldsymbol{w} \in \mathbb{R}^n$ の ℓ_1 ノルムは

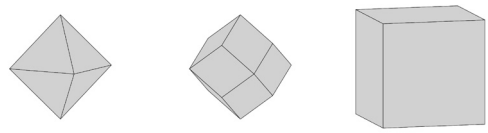
$$\nu(\boldsymbol{w}) = \|\boldsymbol{w}\|_1 := |w_1| + \cdots + |w_n|$$

で与えられる凸関数である．したがって，f が凸関数，S が凸集合であれば (3) および (4) は凸最適化問題であり，局所的最適解を求めれば大域的最適であることが保証される[4]．

(3) ではパラメータ κ を小さくすることで，(4) では η を大きくすることで，得られる解はスパースになる，すなわち，多くのゼロ要素をもつ．特に f を残差平方和とする最小二乗回帰の文脈では，(3) と (4) の定式化は LASSO (Least Absolute Shrinkage and Selection Operator) と呼ばれる．ℓ_1 ノルムの利用がスパース解を得やすくすることの直感的な幾何的説明が可能であるが，(3) あるいは (4) と (2) の差異が意味するものは自明ではないことに注意しておこう．

その他の方法としては

[4] 緩い条件の下，(3) と (4) は等価な問題とみなせる．



$K = 3$ (ℓ_1 ノルム)　　$K = 2$　　$K = 1$ (ℓ_∞ ノルム)

正八面体　　菱形十二面体　　立方体

図 2　$\mathbb{R}^3$ における最大 K ノルム球 $\|\boldsymbol{w}\|_K \le 1$

$$\delta(w) := \begin{cases} 1, & w \neq 0 \text{ のとき}, \\ 0, & w = 0 \text{ のとき}, \end{cases}$$

とすれば，$\|\boldsymbol{w}\|_0 = \sum_{i=1}^n \delta(w_i)$ と書けるため，$\delta(w)$ を近似する連続関数 $\pi(w) \approx \delta(w)$ で置き換えた，$\sum_{i=1}^n \pi(w_i) \approx \|\boldsymbol{w}\|_0$ を最小にする近似手法（たとえば [2, 3]）や，0-1 変数を導入し，0-1 混合整数計画に帰着する方法（たとえば [4]）などさまざまなアプローチが提案されている．本稿では筆者らの研究をベースに，DC 最適化と呼ばれる連続最適化，中でも一次法に分類できる方法に焦点を絞って紹介していく．

2. DC 表現

本節では ℓ_0 制約の等価表現を連続関数により与える方法について述べる．そのために，まずは最大 K ノルムを導入する．

整数 $K \in \{1, \ldots, n\}$ に対し，$\boldsymbol{w} \in \mathbb{R}^n$ の絶対値の意味で大きいほうから K 個の要素の和を**最大 K ノルム**と呼び，$\|\boldsymbol{w}\|_K$ で表す：

$$\|\boldsymbol{w}\|_K := |w_{(1)}| + |w_{(2)}| + \cdots + |w_{(K)}|.$$

ただし $w_{(h)}$ は絶対値で降順に並べたときの h 番目の要素を表す（つまり $|w_{(1)}| \ge |w_{(2)}| \ge \cdots \ge |w_{(n)}|$ である）．$\nu(\boldsymbol{w}) = \|\boldsymbol{w}\|_K$ はノルムであり，$K = n$ のとき ℓ_1 ノルムに等しく ($\|\boldsymbol{w}\|_n = \|\boldsymbol{w}\|_1$)，$K = 1$ のとき ℓ_∞ ノルムに等しい ($\|\boldsymbol{w}\|_1 = \|\boldsymbol{w}\|_\infty := \max_i\{|w_i|\}$)．図 2 に示すように，$\mathbb{R}^3$ 上では $K = 2$ のときノルム球は菱形十二面体と呼ばれる多面体である．

図 2 からも想像がつくように，$\nu(\boldsymbol{w}) = \|\boldsymbol{w}\|_K$ は微分不可能であるが，劣勾配が比較的簡単に求まることが知られている[5]．実際，$\bar{\boldsymbol{w}}$ における $\|\boldsymbol{w}\|_K$ の劣微分

[5] 凸関数 $f : \mathbb{R}^n \to \mathbb{R}$ の $\boldsymbol{w}$ における劣勾配は，任意の $\boldsymbol{z} \in \mathbb{R}^n$ に対して，$f(\boldsymbol{z}) \ge \boldsymbol{s}^\top(\boldsymbol{z} - \boldsymbol{w}) + f(\boldsymbol{w})$ を満たす $\boldsymbol{s}$ として定義される．劣勾配の全体が劣微分である．連続的微分可能な関数における勾配ベクトル $\nabla f(\boldsymbol{w})$ を拡張した概念と言えるが，$-\nabla f(\boldsymbol{w})$ が降下方向になるのに対し，劣勾配を -1 倍した $-\boldsymbol{s}$ は必ずしも降下方向にならないなど，その違いは意外と大きい．

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

は以下で与えられる（たとえば [5, 6]）：

$$\partial \|\bar{\boldsymbol{w}}\|_K = \arg\max_{\boldsymbol{s}} \left\{ \bar{\boldsymbol{w}}^\top \boldsymbol{s} : \begin{array}{l} \sum_{i=1}^{n} |s_i| = K, \\ -\mathbf{1} \le \boldsymbol{s} \le \mathbf{1} \end{array} \right\}. \quad (5)$$

すなわち，劣微分の元である劣勾配を一つ求めるのは，(5) の最適解を一つ求めることに対応している．(5) の最大化問題は，ナップサック問題と同様の構造をもっていることから，以下の手続きにより劣勾配 $\boldsymbol{s}$ が求まる．

1. $\bar{\boldsymbol{w}} = (\bar{w}_1, \ldots, \bar{w}_n)^\top \in \mathbb{R}^n$ を絶対値の降順で並べ替える．その並べ替えに対応する置換を $\sigma : \{1, \ldots, n\} \to \{1, \ldots, n\}$ と置く：
$$|\bar{w}_{\sigma(1)}| \ge |\bar{w}_{\sigma(2)}| \ge \cdots \ge |\bar{w}_{\sigma(n)}|.$$
2. $\bar{\boldsymbol{w}}$ の降順 $\sigma(\cdot)$ に対応して，
$$s_{\sigma(i)} = \begin{cases} \mathrm{sign}(\bar{w}_{\sigma(i)}), & i \in \{1, \ldots, K\} \text{ のとき}, \\ 0, & i \in \{K+1, \ldots, n\} \text{ のとき}, \end{cases}$$
として $\boldsymbol{s} = (s_1, \ldots, s_n)^\top$ を出力する．ただし，
$$\mathrm{sign}(w) := \begin{cases} +1, & w \ge 0, \\ -1. & w < 0. \end{cases}$$

これは各点 $\bar{\boldsymbol{w}}$ における劣勾配が，$\bar{\boldsymbol{w}}$ の要素の絶対値で第 K 番目の要素を見つける程度の手間（理論的には $\mathrm{O}(n)$）で求まることを示唆している．

さて，最大 K ノルムを導入したのはそれを用いることで ℓ_0 制約の等価表現：

$$\|\boldsymbol{w}\|_1 - \|\boldsymbol{w}\|_K = 0 \quad (6)$$

が得られるからである．実際，(1) を満たす $\boldsymbol{w} \in \mathbb{R}^n$ は (6) を満たし，また逆も成り立つ．(6) の左辺を $T_K(\boldsymbol{w})$ と置くと，任意の $\boldsymbol{w} \in \mathbb{R}^n$ に対して

$$T_K(\boldsymbol{w}) = |w_{(K+1)}| + \cdots + |w_{(n)}| \ge 0$$

であり，(6) から，和を構成している各要素がゼロ ($|w_{(K+1)}| = \cdots = |w_{(n)}| = 0$) でなければならない．$\boldsymbol{w}$ の非ゼロ要素数が K 以下であればこれは成り立つし，逆も成り立つので (1) と (6) が等価であることが理解できる．図 3 は $\mathbb{R}^2$ 上で定義した $\|\boldsymbol{w}\|_1, \|\boldsymbol{w}\|_1, T_1(\boldsymbol{w})$ のグラフを示したものである．(iii) から $T_1(\boldsymbol{w}) = 0$ を満たす (w_1, w_2) は w_1 軸と w_2 軸からなる十字形をなすが，これは図 1(ii) で $\|\boldsymbol{w}\|_0 \le 1$ を満たす領域と一致している．

(1) と (6) の等価性を利用すると，(2) は次のように書き換えることができる：

$$\begin{array}{ll} \underset{\boldsymbol{w} \in S}{\text{minimize}} & f(\boldsymbol{w}) \\ \text{subject to} & \|\boldsymbol{w}\|_1 - \|\boldsymbol{w}\|_K = 0. \end{array} \quad (7)$$

(2) が不連続関数である ℓ_0 ノルムを用いて表現している制約を (7) では連続関数 T_K で記述できている．この事実は，(7) を対象とすることで，劣勾配を利用した連続最適化手法の利用可能性を示唆している．

しかしながら，(7) は（連続ながら）依然非凸関数を制約式に含んだ問題であり，このままでは扱いにくい場合が多い．そこで正の定数 ρ を導入し，

$$\underset{\boldsymbol{w} \in S}{\text{minimize}} \quad f(\boldsymbol{w}) + \rho\left(\|\boldsymbol{w}\|_1 - \|\boldsymbol{w}\|_K\right), \quad (8)$$

のように (7) の非凸制約を目的関数に移動した問題を考える．任意の $\boldsymbol{w} \in \mathbb{R}^n$ に対して $T_K(\boldsymbol{w}) \equiv \|\boldsymbol{w}\|_1 - \|\boldsymbol{w}\|_K \ge 0$ であることに注意すると，(8) は元の目的関数 f に加え，(7) の非凸制約の違反も同時に小さくすることを目指していると考えられる．

また，ある条件の下で (8) と (7) は等価，すなわち，十分大きな（有限の）ρ に対する (8) の解は (7)，ひいては (2) の解を与えることを示すことができる．このとき (4) において $\eta = \rho$ とすれば，(4) と (8) との違いは “$-\rho\|\boldsymbol{w}\|_K$” となる．このことは ℓ_0 制約付き最適化問題 (2) と ℓ_1 ノルムを採用したスパース最適化問題による近似とのギャップが，最大 K ノルムによって埋められることを示唆している．

3. DCA

本節では (8) のような問題に対する一次法について紹介する．

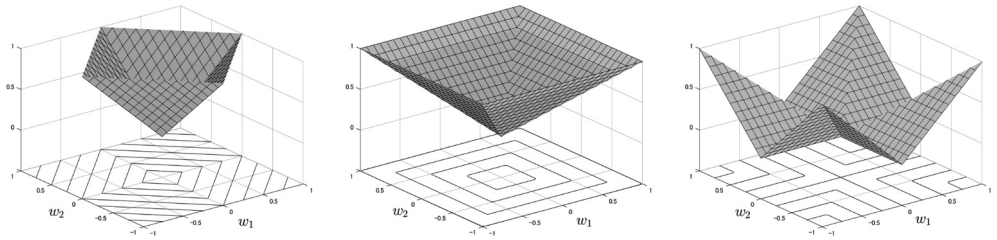
3.1 DC 関数

前節で見た $T_K(\boldsymbol{w}) := \|\boldsymbol{w}\|_1 - \|\boldsymbol{w}\|_K$ のように，二つの凸関数の差 (*D*ifference of two *C*onvex functions) によって表される関数は **DC 関数**と呼ばれる．二つの凸関数の差で表現できなければならないと聞くとかなり限定された関数のクラスに思えるかもしれないが，実は $\mathbb{R}^n$ 上の多項式関数をはじめ，2 回連続微分可能な任意の関数など非常に多くの関数が DC 関数である [7]．また，f の勾配ベクトル ∇f が定数 L のリプシッツ連続，すなわち，$\|\boldsymbol{w}\|_2$ を ℓ_2 ノルム ($\|\boldsymbol{w}\|_2 := \sqrt{w_1^2 + \cdots + w_n^2}$) として，$L > 0$ が存在して，任意の $\boldsymbol{u}, \boldsymbol{v}$ に対し

$$\|\nabla f(\boldsymbol{w}) - \nabla f(\boldsymbol{v})\|_2 \le L\|\boldsymbol{w} - \boldsymbol{v}\|_2 \quad (9)$$

を満たすとする．このとき $\frac{L}{2}\|\boldsymbol{w}\|_2^2 - f(\boldsymbol{x})$ が凸関数になることより，f は

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



(i) $\|\boldsymbol{w}\|_1 = |w_1| + |w_2|$　　(ii) $\|\!|\boldsymbol{w}|\!\|_1 = \max\{|w_1|, |w_2|\}$　　(iii) $T_1(\boldsymbol{w}) := \|\boldsymbol{w}\|_1 - \|\!|\boldsymbol{w}|\!\|_1$

図 3　$\|\boldsymbol{w}\|_1, \|\!|\boldsymbol{w}|\!\|_1, T_1(\boldsymbol{w})$ のグラフ

$$f(\boldsymbol{w}) = \frac{L}{2}\|\boldsymbol{w}\|_2^2 - \Big(\frac{L}{2}\|\boldsymbol{w}\|_2^2 - f(\boldsymbol{x})\Big)$$

と分解できるので，DC である．なお性質 (9) を満たすとき f は L **平滑** (L-smooth) と言う．L 平滑は関数の曲率が一定以下であることを示しており，近接勾配法などの収束性を保証するうえで重要な性質である[6]．

3.2　DCA

DC 関数の最適化に対しては大域的最適性を目指す枠組みもあるが，ここでは局所探索法とみなせる一次法である **DCA** (DC Algorithm) を紹介する[7]．DCA はいわゆる反復法であり，暫定解 $\boldsymbol{w}^{(t-1)}$ から次の暫定解 $\boldsymbol{w}^{(t)}$ を生成するという手続きを，適当な終了条件が満たされるまで繰り返す．ここでは以下の制約なし問題を例に概略を説明する：

$$\underset{\boldsymbol{w}\in\mathbb{R}^n}{\text{minimize}} \quad F(\boldsymbol{w}) := u(\boldsymbol{w}) - v(\boldsymbol{w}). \qquad (10)$$

ここで，u, v ともに凸関数とする．第 $t-1$ 回目の反復終了時に暫定解 $\boldsymbol{w} = \boldsymbol{w}^{(t-1)}$ が得られているとする．第 t 反復においては (10) の暫定解 $\boldsymbol{w}^{(t-1)}$ における関数 v の劣勾配を求め，それを $\boldsymbol{s}^{(t)} \in \partial v(\boldsymbol{w}^{(t-1)})$ とし，次の凸最適化問題 (11) の求解を行い，得られた最適解 $\boldsymbol{w} = \boldsymbol{w}^{(t)}$ を新たな暫定解とする：

$$\underset{\boldsymbol{w}}{\text{minimize}}\ Q(\boldsymbol{w}|\boldsymbol{s}^{(t)}) := u(\boldsymbol{w}) - (\boldsymbol{s}^{(t)})^\top \boldsymbol{w}. \qquad (11)$$

これを適当な終了条件が満たされるまで繰り返すのが DCA である．これをアルゴリズム 1 にまとめる：

アルゴリズム 1：DCA のプロトタイプ

初期解 $\boldsymbol{w}^0$ を与え，$t = 1$ とし，以下の 1. と 2. を適当な終了条件が満たされるまで繰り返す：

repeat

1. $\boldsymbol{s}^{(t)} \in \partial v(\boldsymbol{w}^{(t-1)})$　　［劣勾配計算］
2. $\boldsymbol{w}^{(t)} \in \arg\min_{\boldsymbol{w}} Q(\boldsymbol{w}|\boldsymbol{s}^{(t)})$　　［凸最適化］

($t = t+1$ として 1. へ)

until 適当な終了条件の充足

終了条件としては小さい定数 $\varepsilon > 0$ に対し，$f^{(t-1)} - f^{(t)} < \varepsilon, \|\boldsymbol{w}^{(t)} - \boldsymbol{w}^{(t-1)}\|_2 < \varepsilon$ などが用いられる．

$$V^{(t)} := (\boldsymbol{s}^{(t)})^\top \boldsymbol{w}^{(t-1)} - v(\boldsymbol{w}^{(t-1)})$$

と置くと，

$$\begin{aligned} F(\boldsymbol{w}^{(t-1)}) &= Q(\boldsymbol{w}^{(t-1)}|\boldsymbol{s}^{(t)}) + V^{(t)} \\ &\geq Q(\boldsymbol{w}^{(t)}|\boldsymbol{s}^{(t)}) + V^{(t)} \geq F(\boldsymbol{w}^{(t)}) \end{aligned} \qquad (12)$$

が成り立つ．ここで一つ目の不等号は $\boldsymbol{w}^{(t)}$ が (11) の最適解であることから成り立ち，二つ目の不等号は，劣勾配の定義より，任意の $\boldsymbol{w} \in \mathbb{R}^n$ に対して

$$-v(\boldsymbol{w}) \leq -(\boldsymbol{s}^{(t)})^\top(\boldsymbol{w} - \boldsymbol{w}^{(t-1)}) - v(\boldsymbol{w}^{(t-1)})$$

が成り立つことによる．(12) より，DCA は反復ごとに目的関数値 $F(\boldsymbol{w}^{(t)})$ が単調に改善するように（正確には「増加することなく」）点列 $\{\boldsymbol{w}^{(t)} : t = 0, 1, \dots\}$ を生成していく[8]．この生成点列は $F(\boldsymbol{w}) = u(\boldsymbol{w}) - v(\boldsymbol{w})$ の**臨界点** (critical point)，すなわち

$$\boldsymbol{0} \in \partial u(\boldsymbol{w}) - \partial v(\boldsymbol{w})$$

なる $\boldsymbol{w}$ への収束が保証される [9, 10]．

3.3　近接 DCA

前項で述べた DCA のプロトタイプでは各反復で凸

[6] 近接勾配法については [1] の第 3 節や [8] の第 5 節を参照いただきたい．

[7] 機械学習分野では CCCP (Convex-ConCave Procedure) などとも呼ばれる．

[8] このように各反復で上界関数を最小化するアルゴリズムは**上界最小化アルゴリズム**（**MM アルゴリズム**）と呼ばれる．

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

最適化問題 (11) を解くことが前提となっている．しかし大規模な問題に対しては，各反復で別の最適化アルゴリズムを呼び出すことは解法の実用性を損なう．そこで問題の構造を活かし，各反復での計算を軽くする工夫が考えられる．

凸関数 f, g, h により

$$\underset{\boldsymbol{w}\in\mathbb{R}^n}{\text{minimize}} \quad F(\boldsymbol{w}) = f(\boldsymbol{w}) + g(\boldsymbol{w}) - h(\boldsymbol{w}), \tag{13}$$

で表される問題を考える．たとえば $g(\boldsymbol{w}) = \rho\|\boldsymbol{w}\|_1$, $h(\boldsymbol{w}) = \rho\|\boldsymbol{w}\|_K$, $S = \mathbb{R}^n$ とすれば問題 (8) に一致する．さらに f が L 平滑な凸，g が近接写像が利用できる凸関数とする．g の近接写像とは以下で定義される写像である：

$$\mathrm{prox}_g(\boldsymbol{z}) := \arg\min_{\boldsymbol{x}} \Big\{ g(\boldsymbol{x}) + \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{z}\|_2^2 \Big\}.$$

ここでは簡単な演算により関数 g の近接写像が評価できることを仮定しており，最適化計算をいちいち行う必要はない．

g, h の凸性および f の L 平滑性から，F は

$$F(\boldsymbol{w}) = \Big(\frac{L}{2}\|\boldsymbol{w}\|_2^2 + g(\boldsymbol{x})\Big) - \Big(\frac{L}{2}\|\boldsymbol{w}\|_2^2 - f(\boldsymbol{w}) + h(\boldsymbol{w})\Big)$$

と DC 分解できる．ここで f について前節で見た L 平滑関数の DC 分解を利用していることに注意する．$\bar{\boldsymbol{w}} := \boldsymbol{w}^{(t-1)}$ における h の劣勾配を $\bar{\boldsymbol{s}}$ と表せば，

$$\begin{aligned}
\arg\min_{\boldsymbol{w}} Q(\boldsymbol{w} \mid \bar{\boldsymbol{s}}) &= \arg\min_{\boldsymbol{w}} \Big\{ \frac{L}{2}\|\boldsymbol{w} - \bar{\boldsymbol{w}}\|_2^2 + g(\boldsymbol{w}) \\
&\qquad + (\nabla f(\bar{\boldsymbol{w}}) - \bar{\boldsymbol{s}})^\top \boldsymbol{w} \Big\} \\
&= \arg\min \Big\{ \frac{1}{L} g(\boldsymbol{w}) \\
&\qquad + \frac{1}{2}\|\boldsymbol{w} - \Big[\bar{\boldsymbol{w}} - \frac{1}{L}(\nabla f(\bar{\boldsymbol{w}}) - \bar{\boldsymbol{s}})\Big]\|_2^2 \Big\} \\
&= \mathrm{prox}_{g/L}\Big(\bar{\boldsymbol{w}} - \frac{1}{L}\big(\nabla f(\bar{\boldsymbol{w}}) - \bar{\boldsymbol{s}}\big)\Big),
\end{aligned}$$

となる．したがって，アルゴリズム 1 のステップ 2 は

$$\boldsymbol{w}^{(t)} = \mathrm{prox}_{g/L}\Big(\boldsymbol{w}^{(t-1)} - \frac{1}{L}(\nabla f(\boldsymbol{w}^{(t-1)}) - \boldsymbol{s}^{(t)})\Big)$$

のように変形でき，g の近接写像が陽に与えられる場合には効率的に計算可能となる．実際 (8) であれば，$g(\boldsymbol{w}) = \rho\|\boldsymbol{w}\|_1$ であるから

$$\boldsymbol{y} := \boldsymbol{w}^{(t-1)} - \frac{1}{L}\big(\nabla f(\boldsymbol{w}^{(t-1)}) - \boldsymbol{s}^{(t)}\big)$$

として近接写像は以下のように与えられる：

$$\begin{aligned}
\mathrm{prox}_{g/L}(\boldsymbol{y}) &= \mathrm{prox}_{(\rho/L)\|\cdot\|_1}(\boldsymbol{y}) \\
&= (\mathrm{soft}_{\rho/L}(y_1), \ldots, \mathrm{soft}_{\rho/L}(y_n)).
\end{aligned}$$

ここで soft_C は軟閾値演算 (soft-thresholding) であり，定数 $C > 0$ に対して以下で定義される：

$$\mathrm{soft}_C(y) := \begin{cases} y + C, & y \le -C, \\ 0, & -C \le y \le C, \\ y - C, & y \ge C. \end{cases} \tag{14}$$

この場合のように近接写像が簡単に計算できる場合には，各反復で凸最適化問題を解く必要はない．

このように近接勾配法と組み合わせた DCA は**近接 DCA** (PDCA) と呼ばれる [11]．加えて h の劣勾配が簡単に計算できれば，(13) に対する PDCA の 1 反復は効率的に行うことができる．したがって，f が L 平滑なとき，(8) に対する PDCA は最大 K ノルムの劣勾配計算と軟閾値演算の反復計算に帰着されることから，大規模な問題に対しても効率的な適用が可能となる．

3.4 高速化技法

PDCA は各反復の計算は効率的であるが，生成点列の収束という意味では工夫の余地が残る．たとえば，近接勾配法と同様に PDCA ではステップサイズが L 平滑関数 f を特徴づける L の逆数に固定されている．したがって，L が大きい場合，暫定解の更新が小さくなってしまう．実際にはステップサイズは目的関数を改善する範囲でなるべく大きくとることが好ましく，バックトラッキングと呼ばれる方法が実用的とされる．Tono et al. [12] は PDCA にその工夫を取り入れた方法および必ずしも目的関数が単調に改善しない（非単調な）更新を許した拡張も提示している．

一方，Wen et al. [13] は Nesterov の外挿を施した改良版である PDCA_e を提示している．ステップ 1, 3 は PDCA の 1 と 2 に対応するが，その間に外挿と呼ば

アルゴリズム 2： PDCA_e

$\sup_t \beta^{(t)} < 1$ なる $\{\beta^{(t)}\} \subset [0, 1)$ を定め，$\boldsymbol{w}^{(-1)} = \boldsymbol{w}^{(0)}$, $t = 1$ とし，以下を繰り返す：

repeat

1. $\boldsymbol{s}^{(t)} \in \partial h(\boldsymbol{w}^{(t-1)})$
2. $\boldsymbol{y}^{(t)} = \boldsymbol{w}^{(t-1)} + \beta^{(t)}(\boldsymbol{w}^{(t-1)} - \boldsymbol{w}^{(t-2)})$
3. $\boldsymbol{w}^{(t)} = \underset{\boldsymbol{y}}{\mathrm{argmin}} \Big\{ (\nabla f(\boldsymbol{y}^{(t)}) - \boldsymbol{s}^{(t)})^\top \boldsymbol{y} + \frac{L}{2}\|\boldsymbol{y} - \boldsymbol{y}^{(t)}\|_2^2 + g(\boldsymbol{y}) \Big\}$

（$t = t + 1$ として 1. へ）

until 適当な終了条件の充足

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

れるステップ 2 が入る形になっている．$\beta^{(t)}=0$ とすれば，PDCA に一致する．そうでない場合で，もし問題が凸最適化（すなわち $h=0$）であれば，いわゆる加速化が理論的に保証されている．[13] は，この外挿を組み入れることで加速効果が見られることを指摘している．ちなみに [12] では [13] とも比較し，[12] の手法の方が効率的に加速できる例を示している．

また，各反復の勾配計算を軽くするために用いられ，近年盛んに研究・適用されている確率的勾配法のアイデアを DCA に用いる手法なども提案されており，その有効性が主張されている [14]．概要については本特集記事 [15] を参照していただきたい．

3.5 EDCA

Pang et al. [16] は (13) の目的関数の h が I 個の凸関数 $\psi_1,\dots,\psi_I$ の最大値，すなわち

$$h(\boldsymbol{w})=\max\{\psi_1(\boldsymbol{w}),\dots,\psi_I(\boldsymbol{w})\}$$

と表される場合に，臨界点よりも少し強い概念である**方向停留点** (directional-stationary point)，すなわち，任意の方向 $\boldsymbol{d}\in\mathbb{R}^n$ に対して方向微分が非負，

$$F'(\boldsymbol{w};\boldsymbol{d}):=\lim_{t\searrow 0}\frac{F(\boldsymbol{w}+t\boldsymbol{d})-F(\boldsymbol{w})}{t}\geq 0\,,\quad(15)$$

を満たす $\boldsymbol{w}$ を求めるアルゴリズムとして Enhanced DCA (EDCA) を提案している．(15) は

$$\partial v(\boldsymbol{w})\subset\partial u(\boldsymbol{w})$$

と等価であることから，方向停留点は臨界点であり，逆に臨界点であっても方向停留点でない例があるという意味で，より強い最適性条件となっている．EDCA を適用すれば方向停留点を得ることが保証される一方，手間がかかる．Lu et al. は [17] で EDCA に近接勾配法の要素を入れた EPDCA を提示し，[18] では非単調な線形探索や確率的な工夫を取り入れることで，効率的に方向停留点を求めるアルゴリズム (NEPDCA) を提案している．[18] によれば，そういった工夫を取り入れることで，臨界点への収束しか保証されていない PDCA_e よりも質のよい解が効率よく求まる．

4. T_K の近接写像と ADMM

この節では DCA を離れ，Bertsimas et al. の論文 [19] で提示されている**交互方向乗数法 (Alternating Direction Method of Multiplier; ADMM)** による (8) に対するアプローチを紹介する．

すでに見たように $T_K(\boldsymbol{w})$ は非凸な関数であるが，[19] ではその近接写像（先の 1 点）が簡単に計算できることを指摘し，それを利用して基数制約付き LASSO 回帰への適用を考えている．具体的には $S=\mathbb{R}^n$ のときの (8) において，f を以下で与えた場合を考えている：

$$f(\boldsymbol{w})=\sum_{i=1}^m(y_i-\boldsymbol{w}^\top\boldsymbol{x}_i)^2+\eta\|\boldsymbol{w}\|_1.$$

このとき (8) が (16) のように書けることに注意する：

$$\begin{array}{ll}\underset{\boldsymbol{w},\boldsymbol{z}}{\text{minimize}} & f(\boldsymbol{w})+\rho T_K(\boldsymbol{z}),\\ \text{subject to} & \boldsymbol{w}=\boldsymbol{z}.\end{array}\quad(16)$$

制約条件 $\boldsymbol{w}-\boldsymbol{z}=\boldsymbol{0}$ に対するラグランジュ乗数を $\boldsymbol{\lambda}$,$\tau>0$ を定数とし，(16) の拡張ラグランジュ関数を

$$\begin{aligned}L_\tau(\boldsymbol{w},\boldsymbol{z},\boldsymbol{\lambda}):=\ & f(\boldsymbol{w})+\rho T_K(\boldsymbol{z})\\ &+\boldsymbol{\lambda}^\top(\boldsymbol{w}-\boldsymbol{z})+\frac{\tau}{2}\|\boldsymbol{w}-\boldsymbol{z}\|_2^2\end{aligned}$$

と置く．ADMM の詳細については本特集記事 [1] もしくは網羅的なチュートリアル [20] を参照していただきたいが，(16) に対する ADMM の手続きは，アルゴリズム 3 に示すように，各反復において $L_\tau(\boldsymbol{w},\boldsymbol{z},\boldsymbol{\lambda})$ の最小化をステップ 1 で $\boldsymbol{w}$，ステップ 2 で $\boldsymbol{z}$ についてそれぞれ分けて行い，ステップ 3 で $\boldsymbol{\lambda}$ の更新を行うというものである．

アルゴリズム 3：(16) に対する ADMM

$\boldsymbol{z}^{(0)}$, $\boldsymbol{\lambda}^{(0)}$, $\tau>0$ を与え，$t=1$ とする．

repeat

1. $\boldsymbol{w}^{(t)}=\arg\min_{\boldsymbol{w}} L_\tau(\boldsymbol{w},\boldsymbol{z}^{(t-1)},\boldsymbol{\lambda}^{(t-1)})$
2. $\boldsymbol{z}^{(t)}\in\arg\min_{\boldsymbol{z}} L_\tau(\boldsymbol{w}^{(t)},\boldsymbol{z},\boldsymbol{\lambda}^{(t-1)})$
   $=\mathrm{prox}_{T_K/\tau}\left(\boldsymbol{w}^{(t)}+\frac{1}{\tau}\boldsymbol{\lambda}^{(t-1)}\right)$
3. $\boldsymbol{\lambda}^{(t)}=\boldsymbol{\lambda}^{(t-1)}+\tau(\boldsymbol{x}^{(t)}-\boldsymbol{z}^{(t)})$
   （$t=t+1$ として 1. へ）

until 適当な終了条件の充足

まず，ステップ 2 が近接写像の形で与えられるのがポイントである．また [19, 21] によれば，ステップ 2 の近接写像（の一つ）はベクトル要素の並べ替えと解析的な演算により求めることができる．実際 $\boldsymbol{y}^{(t)}:=\boldsymbol{x}^{(t)}+\frac{1}{\tau}\boldsymbol{\lambda}^{(t-1)}$ とし，その降順による並べ替えの結果得られる上位 K 個の添え字集合を $I^{(t)}$ とすると，近接写像（の一つ）は

$$z_i^{(t)}=\begin{cases}y_i^{(t)}, & i\in I^{(t)},\\ \mathrm{soft}_{1/\tau}(y_i^{(t)}), & i\notin I^{(t)},\end{cases}$$

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

で得られる．ただし $\mathrm{soft}_{1/\tau}(\cdot)$ は (14) で与えられる軟閾値演算である．[19] ではステップ 1 の凸最適化計算を近似的に解くことを行っているようであるが，ADMM のほうが DCA よりもよい解に収束するという計算結果を示している．一方 [22] では f が線形方程式の指標関数 (indicator function) で与えられるケースを考え，ステップ 1, 2 とも近接写像になるような場合を考え，効率的なアルゴリズムを提示している．このように，関数 T_K の近接写像が簡単に計算できることを利用することで，ADMM や近接勾配法のようなアルゴリズムの利点が大きくなる．

5. 階数制約への拡張

2 節で紹介した実ベクトルの ℓ_0 制約およびその DC 表現は，行列変数の階数制約にも応用することができる．つまり，実行列 $\boldsymbol{W} \in \mathbb{R}^{m\times n}$ および $1 \leq K < \min\{m,n\}$ なる自然数 K に対し，その行列の階数 $\mathrm{rank}(\boldsymbol{W})$ が K 以下であるという条件を考える：

$$\mathrm{rank}(\boldsymbol{W}) \leq K. \tag{17}$$

このような制約はさまざまな応用で現れるが，データ解析における有名な例として推薦システムにおけるユーザー・アイテム行列の補完などがある．ただし，この制約は最適化問題にある種の非凸性をもたらすため，階数の凸近似として，特異値の和として定義される行列の核ノルム $\|\boldsymbol{W}\|_*$ を制限する凸最適化問題を解くのが代表的である．核ノルムはベクトルの ℓ_1 ノルムに相当する行列ノルムである．

一方で階数は非ゼロな特異値の個数に等しいという事実を用いると，次のように階数制約 (17) の等価表現が得られる [11].

$$\|\boldsymbol{W}\|_* - \|\boldsymbol{W}\|_K = 0$$

ここで $\|\boldsymbol{W}\|_K$ は $\boldsymbol{W}$ の特異値のうち大きいほうから K 個の和を表し，**Ky Fan K ノルム**と呼ばれる．すなわち $\sigma_i(\boldsymbol{W})$ を i 番目に大きい $\boldsymbol{W}$ の特異値として，

$$\|\boldsymbol{W}\|_K := \sum_{i=1}^{K} \sigma_i(\boldsymbol{W})$$

である．Ky Fan K ノルムについても劣勾配は特異値計算から求めることができるため，ベクトルの場合と同様に DC 最適化によるアプローチが適用可能である．

6. まとめ

本稿では不連続関数を用いて捉えられることが多い ℓ_0 制約に対して，連続関数を用いた等価な置き換えと，得られた最適化問題に対する一次法のいくつかを紹介した．等価な置き換えについては二つのノルムの差に基づく方法のみを紹介したが，もう少し一般的な形で表現することもできる [12, 21, 22].

また，問題が非凸な構造をもつため，一次法に基づく局所探索のみでは大域的最適性を保証できない．実際 [19] の計算実験などでも 0-1 混合整数計画による大域的最適解からの乖離が見られる．本問題に限らず，求解の効率性と解の質との折り合いをどうつけていくかは非凸最適化の応用に附随する普遍的課題と言える．

謝辞　本稿は科学研究費基盤研究 (C)15K01204 の補助を部分的に受けた研究に基づいている．

参考文献

[1] 小野峻佑，“近接分離アルゴリズムとその応用—信号処理・画像処理的観点から—，”オペレーションズ・リサーチ：経営の科学，**64**(6), pp. 316–325, 2019.
[2] J. Fan and R. Li, “Variable selection via nonconcave penalized likelihood and its oracle properties,” *Journal of the American Statistical Association*, **96**, pp. 1348–1360, 2001.
[3] C. H. Zhang, “Nearly unbiased variable selection under minimax concave penalty,” *The Annals of Statistics*, pp. 894–942, 2010.
[4] T. S. Arthanari and Y. Dodge, *Mathematical Programming in Statistics*, John Wiley & Sons, 1981.
[5] G. A. Watson, “On matrix approximation problems with Ky Fan k norms,” *Numerical Algorithms*, **5**, pp. 263–272, 1993.
[6] B. Wu, C. Ding, D. F. Sun and K. C. Toh, “On the Moreau–Yoshida regularization of the vector k-norm related functions,” *SIAM Journal on Optimization*, **24**, pp. 766–794, 2014.
[7] H. Tuy, *Convex Analysis and Global Optimization*, 2nd edition, Springer, 2016.
[8] 伊藤勝，“凸最適化問題に対する一次法とその理論—加速勾配法とその周辺—，”オペレーションズ・リサーチ：経営の科学，**64**(6), pp. 326–334, 2019.
[9] T. Pham Dinh and H. A. Le Thi, “Convex analysis approach to d.c. programming: Theory, algorithms and applications,” *Acta Mathematica Vietnamica*, **22**, pp. 289–355, 1997.
[10] T. Pham Dinh and H. A. Le Thi, “Recent advances in DC programming and DCA,” *Transactions on Computational Collective Intelligence*, **8342**, pp. 1–37, 2014.
[11] J. Gotoh, A. Takeda and K. Tono, “DC formulations and algorithms for sparse optimization problems,” *Mathematical Programming*, **169**, pp. 141–176, 2018.
[12] K. Tono, A. Takeda and J. Gotoh, “Efficient DC algorithm for constrained sparse optimization,” arXiv: 1701.08498, 2017.
[13] B. Wen, X. Chen and T. K. Pong, “A proximal difference-of-convex algorithm with extrapolation,”

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

Computational Optimization and Applications, **69**, pp. 297–324, 2018.
[14] A. Nitanda and T. Suzuki, "Stochastic difference of convex algorithm and its application to training deep Boltzmann machine," In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pp. 470–478, 2017.
[15] 鈴木大慈，二反田篤史，村田智也，"機械学習問題における確率的最適化技法，"オペレーションズ・リサーチ：経営の科学，**64**(6), pp. 360–367, 2019.
[16] J.-S. Pang, M. Razaviyayn and A. Alvarado, "Computing B-stationary points of nonsmooth DC programs," *Mathematics of Operations Research*, **42**, pp. 95–118, 2016.
[17] Z. Lu, Z. Zhou and Z. Sun, "Enhanced proximal DC algorithms with extrapolation for a class of structured nonsmooth DC minimization," *Mathematical Programming*, https://doi.org/10.1007/s10107-018-1318-9, 2018.
[18] Z. Lu, Z. Zhou and Z. Sun, "Enhanced proximal DC algorithms for a class of structured nonsmooth DC minimization," technical report, http://www.math.hkbu.edu.hk/~zirui-zhou/papers/nsmth-dc.pdf（2019 年 3 月 1 日閲覧）
[19] D. Bertsimas, M. S. Copenhaver and R. Mazumder, "The Trimmed Lasso: Sparsity and Robustness," arXiv: 1708.04527, 2017.
[20] S. Boyd, N. Parikh, E. Chu, B. Peleato and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, **3**, pp. 1–122, 2011.
[21] Z. Lu and X. Li, "Sparse recovery via partial regularization: Models, theory and algorithms," *Mathematics of Operations Research*, **43**, pp. 1290–1316, 2018.
[22] 後藤順哉，福田琢巳，"ノイズのない場合の k-疎復元に対する刈込 ℓ_p 関数を用いた定式化と ADMM の適用，"日本オペレーションズ・リサーチ学会春季研究発表会アブストラクト集，pp. 236–237, 2019.

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