Vol. 37 (2017)
No. 6

数 学 杂 志
J. of Math. (PRC)

带 1 - 范数约束的分裂可行问题的投影算法

畅含笑, 屈 彪
(曲阜师范大学管理学院, 山东 日照 276826)

摘 要: 本文主要研究带 1 - 范数约束的分裂可行问题的求解算法. 用一种交替投影算法, 求得了问题的解, 提出松弛交替投影算法, 改进了直接往闭凸集上投影这一不足, 并证明了该算法的收敛性.

关键词: 分裂可行问题; 1 - 范数约束; 交替投影; 松弛
MR(2010) **主题分类号**: 90C25; 65K05 **中图分类号**: O221.2
文献标识码: A **文章编号**: 0255-7797(2017)06-1234-11

1 引言

由于社会的发展和生产实践的需求, 最优化问题在金融、医学、药学以及手工农业等领域受到越来越多的重视, 分裂可行问题就是其中之一. 1994 年, Censor 和 Elfving 依据医学中有关放射治疗的实践经验和理论提出了分裂可行问题 [1], 问题形式如下

$$\text{求}\ \ x \in C \ \text{ s.t. }\ Ax \in Q, \tag{1.1}$$

其中 C, Q 分别为 R^N 和 R^M 中的非空闭凸集, A 为一个 $M \times N$ 的实矩阵.

随后, 分裂可行问题的拓展形式被学者们相继提出. 本文主要研究的是带 1 - 范数约束的分裂可行问题的解, 问题形式如下

$$\text{求}\ \ x \in C, Ax \in Q \ \text{ s.t. }\ \|x\|_1 \le s, \tag{1.2}$$

其中 $\|.\|_1$ 代表 1 - 范数, $s > 0$ 为一个给定的常数.

目前, 大多数关于求解分裂可行问题及其拓展问题的算法, 要么牵涉到往闭凸集上的投影, 而这一投影在实际操作中难以实现; 要么在求解合适的步长过程中需要估算相关矩阵的最大特征值、Lipschitz 系数或进行线搜索, 而这些操作往往需要大量的计算. 关于求解分裂可行问题的 1 - 范数解问题已有学者在研究 [14,15]. 在文献 [14] 中, 作者将分裂可行问题的 1 - 范数解问题转化为变分不等式问题进行求解, 但在其给出的算法中需要估算矩阵的最大特征值.

在文献 [2] 中, 对于多集分裂可行问题

$$x \in \bigcap_{i=1}^{t} C_i \ \text{ s.t. }\ Ax \in \bigcap_{j=1}^{r} Q_j, \tag{1.3}$$

其中 $C_i\ (i = 1, 2, \cdots, t)$ 和 $Q_j\ (j = 1, 2, \cdots, r)$ 分别为 R^N 和 R^M 中的非空闭凸集.

*收稿日期: 2016-03-11 接收日期: 2016-06-28
基金项目: 国家自然科学基金 (11271226); 山东省优秀中青年科学家科研奖励基金 (BS2012SF027).
作者简介: 畅含笑 (1990–), 女, 河南开封, 硕士, 主要研究方向: 非线性规划.

作者提出了一种序列投影算法

$$x^{k+1} = P_{C_t} P_{C_{t-1}} \cdots P_1(x^k + \omega_k r_k d_k),$$

其中 $d_j^k = P_{Q_j}(Ax_k) - Ax_k, j = 1, 2, \cdots r, d_k = \sum\limits_{j=1}^{r} A^T d_j^k, r_k = \dfrac{\sum\limits_{j=1}^{r} \|d_j^k\|^2}{\|d_k\|^2}, 0 < \underline{\omega} \leq \omega_k \leq \overline{\omega} < 2, k = 1, 2, \cdots$.

显然, 序列投影算法有可以直接计算的步长, 从而避免了在求解步长时进行大量的计算. 受到文献 [2] 的启发, 将问题 (1.2) 转化为如下形式

$$\text{求} \ \ x \in C \cap D \ \ \text{s.t.} \ \ Ax \in Q, \tag{1.4}$$

其中 $D = \{x | \|x\|_1 \leq s\}$.

序列投影算法虽有可以直接计算的步长, 但却牵涉到往闭凸集上的投影. 首先, 本文在序列投影的基础上, 提出交替投影算法求解了问题 (1.4). 其次, 考虑到在实际操作中往闭凸集上的投影是难以实现的, 文章后半部分构造包含 C, D, Q 的半空间, 利用到半空间上的投影来代替到闭凸集上的投影, 从而使投影得到了顺利实现.

2 预备知识

定义 1 [3,4] 设 $\Omega \subset R^N$ 为非空闭凸集, 对任意的 $x \in R^N$, 定义

$$P_\Omega(x) = \arg\min\{\|x - y\| | y \in \Omega\}.$$

并称其为 x 到 Ω 上的投影. 显然, 若 $x \in \Omega$, 则 $x = P_\Omega(x)$.

注 1 由投影的定义, 可以得出问题 (1.2) 等价于如下带约束的最小化问题

$$\begin{aligned} &\min \ \ \frac{1}{2}\|Ax - P_Q(Ax)\|^2, \\ &\text{s.t.} \ \ x \in C \cap D. \end{aligned}$$

定义 2 [3] 给定正常凸函数 $f : R^N \to (-\infty, +\infty]$, 对任意一点 $x \in \text{dom}f$, 若向量 $\xi \in R^N$ 满足

$$f(y) - f(x) \geq \langle \xi, y - x \rangle, y \in R^N, \tag{2.1}$$

则称 ξ 为凸函数 f 在点 x 处的次梯度. 将满足式 (2.1) 的向量 ξ 的全体构成的集合记为 $\partial f(x)$, 并称之为 f 在 x 处的次微分.

注 2 $\text{dom}f$ 表示 f 的有效域, 即 $\text{dom}f = \{x \in R^N | f(x) < +\infty\}$. 若 f 在 x 处可微, 则满足 (2.1) 式的 ξ 存在且唯一, 并且易知该向量即梯度 $\nabla f(x)$.

由次梯度的定义, 简单计算可得 $f(x) = \|x\|_1$ 在 $x \in R^N$ 的次梯度 t, 其分量为

$$t_j = \begin{cases} 1, & x_j > 0; \\ [-1,1], & x_j = 0; \\ -1, & x_j < 0. \end{cases}$$

引理 2.1 [9] 设 Ω 为 R^N 中的非空闭凸集, 则对于投影算子 $P_\Omega(.)$, 下述结论成立.

(a) $\langle P_\Omega(x)-x, z-P_\Omega(x)\rangle \geq 0, \forall x \in R^N, z \in \Omega$;

(b) $\|P_\Omega(x)-P_\Omega(y)\| \leq \|x-y\|, \forall x, y \in R^N$;

(c) $\|P_\Omega(x)-z\|^2 \leq \|x-z\|^2-\|P_\Omega(x)-x\|^2, \forall x \in R^N, z \in \Omega$;

(d) $\|P_\Omega(x)-P_\Omega(y)\|^2 \leq \|x-y\|^2-\|P_\Omega(x)-x+y-P_\Omega(y)\|^2, \forall x, y \in R^N$.

引理 2.2 [3] 假设 $f: R^N \to R$ 为凸函数, 则 f 在 R^N 上处处次可微且次微分在 R^N 的任意有界子集上一致有界.

引理 2.3 [4] 设 $a \in R^N, b \in R$, 则 $u \in R^N$ 到半空间 $\{x \in R^N | a^T x \geq b\}$ 上的投影为

$$P(u)=u+\frac{\max\{0, b-a^T u\}}{\|a\|^2} a.$$

引理 2.4 [16] 设 $f: R^N \to (-\infty, +\infty]$ 为闭正常凸函数. 给定满足 $x^k \to x$ 的点列 $\{x^k\} \subseteq R^N$, 若 $\xi^k \in \partial f(x^k)$ 且 $\xi^k \to \xi$, 则有 $\xi \in \partial f(x)$.

3 算法及收敛性分析

此部分给出问题 (1.4) 的求解算法.

3.1 交替投影算法

本文采用可以直接计算的步长, 避免了在求合适步长时, 进行大量的计算. 在该算法中, 假设到闭凸集上的投影是可以直接计算的, 且问题 (1.4) 的解集非空.

算法 3.1

步骤 1 任取 $x^0 \in R^N$, 选取 $0 \leq \underline{\omega} \leq \omega_k \leq \bar{\omega} < 2, k=0,1,2,\cdots$;

步骤 2 令 $\overline{d_k}=P_Q(Ax_k)-Ax_k, d_k=A^T\overline{d_k}, r_k=\dfrac{\|\overline{d_k}\|^2}{\|d_k\|^2}$;

步骤 3 计算 $x^{k+1}=P_D P_C(x^k+\omega_k r_k d_k)$.

引理 3.1 设 x^* 为问题 (1.4) 的任意解, $\{x^k\}$ 为算法 3.1 产生的点列, 则数列 $\{\|x^k-x^*\|\}$ 收敛.

定理 3.1 设 $\{x^k\}$ 为算法 3.1 产生的点列, 则 $\{x^k\}$ 收敛到问题 (1.4) 的一个解.

注 3 算法 3.1 可以看成 [2] 中序列投影算法的特例, 引理 3.1、定理 3.1 的证明过程可分别参看文献 [2] 中引理 2.2、定理 2.1.

3.2 松弛交替投影算法

由于算法 3.1 牵涉到往闭凸集上的投影, 所以针对这一不足对算法 3.1 进行改进. 这里, 寻找包含集合 C, D, Q 的半空间, 利用到半空间上的投影来代替到闭凸集上的投影, 从而使投影简单可行.

在该部分, 将集合 $D=\{x \in R^N | \|x\|_1 \leq s\}$ 记为 $D=\{x \in R^N | f(x) \leq 0\}$, 其中 $f(x)=\|x\|_1-s$.

假设 C, Q 满足如下条件:

(H1) $C=\{x \in R^N | c(x) \leq 0\}$, 其中 $c: R^N \to R$ 是凸函数 (不需要可微的) 且 C 是非空的. $Q=\{y \in R^M | q(y) \leq 0\}$, 其中 $q: R^M \to R$ 是凸函数 (不需要可微的) 且 Q 是非空的.

(H2) 对于任意的 $x \in R^N$, 至少有一个次梯度 $\varsigma \in \partial c(x)$ 是可以计算的, 其中 $\partial c(x) = \{\varsigma \in R^N | c(z) \geq c(x) + \langle \varsigma, z-x \rangle, \forall z \in R^N\}$.

对于任意的 $y \in R^M$, 至少有一个次梯度 $\eta \in \partial q(y)$ 是可以计算的, 其中 $\partial q(y) = \{\eta \in R^M | q(u) \geq q(y) + \langle \eta, u-y \rangle, \forall u \in R^M\}$. 并且构造包含 C, D, Q 的半空间 C_k, D_k, Q_k 如下

$$C_k = \{x \in R^N | c\left(x^k\right) + \left\langle \varsigma^k, x - x^k \right\rangle \leq 0\}, \text{ 其中 } \varsigma^k \in \partial c\left(x^k\right).$$
$$D_k = \{x \in R^N | f\left(x^k\right) + \left\langle \phi^k, x - x^k \right\rangle \leq 0\}, \text{ 其中 } \phi^k \in \partial f\left(x^k\right).$$
$$Q_k = \{y \in R^M | q\left(Ax^k\right) + \left\langle \eta^k, y - A x^k \right\rangle \leq 0\}, \text{ 其中 } \eta^k \in \partial q\left(Ax^k\right).$$

注 4 由次梯度的定义, 容易得出 $\forall k \in N$, 有 $C \subset C_k, D \subset D_k, Q \subset Q_k$.

算法 3.2

步骤 1 任取 $x^0 \in R^N$, 选取 $0 < \underline{w} \leq w_k \leq \overline{w} < 2, k = 0, 1, 2, \cdots$;

步骤 2 令 $\overline{d_k} = P_{Q_k}(Ax_k) - Ax_k, d_k = A^T \overline{d_k}, r_k = \dfrac{\left\|\overline{d_k}\right\|^2}{\left\|d_k\right\|^2}$;

步骤 3 计算 $x^{k+1} = P_{D_k} P_{C_k}\left(x^k + w_k r_k d_k\right)$.

其中 C_k, D_k, Q_k 如上文所述.

注 5 在实际计算中,

$$\overline{d_k} = P_{Q_k}(Ax_k) - Ax_k = \frac{\max\{0, q(Ax^k)\}}{\|\eta^k\|^2}(-\eta^k),$$
$$r_k = \frac{\left\|\overline{d_k}\right\|^2}{\left\|d_k\right\|^2} = \frac{\|P_{Q_k}(Ax_k) - Ax_k\|^2}{\|A^T(P_{Q_k}(Ax_k) - Ax_k)\|^2} = \frac{\|\frac{\max\{0, q(Ax^k)\}}{\|\eta^k\|^2}\eta^k\|^2}{\|A^T(\frac{\max\{0, q(Ax^k)\}}{\|\eta^k\|^2}\eta^k)\|^2},$$
$$x^{k+1} = P_{D_k} P_{C_k}\left(x^k + w_k r_k d_k\right) = P_{D_k}(x^k + \omega_k r_k d_k + \frac{\max\{0, c(x^k) + (\varsigma^k)^T(\omega_k r_k d_k)\}}{\|\varsigma^k\|^2}(-\varsigma^k)).$$

令

$$y = x^k + \omega_k r_k d_k + \frac{\max\{0, c(x^k) + (\varsigma^k)^T(\omega_k r_k d_k)\}}{\|\varsigma^k\|^2}(-\varsigma^k),$$

则

$$x^{k+1} = P_{D_k} P_{C_k}\left(x^k + w_k r_k d_k\right) = P_{D_k}(y) = y + \frac{\max\{0, f(x^k) - (\phi^k)^T x^k + (\phi^k)^T y\}}{\|\phi^k\|^2}(-\phi^k).$$

对任意的 $\tilde{x} \in R^N$, 记

$$C(\tilde{x}) = \{x \in R^N | c(\tilde{x}) + \langle \tilde{\varsigma}, x - \tilde{x} \rangle \leq 0\}, \tag{3.1}$$

其中 $\tilde{\varsigma} \in \partial c(\tilde{x})$.

注 6 由 (3.1) 式及 C_k 的定义, 显然 $C_k = C(x^k)$. 在式 (3.1) 中, 可以取满足如下条件的 $\tilde{x}, \tilde{\varsigma}: \lim\limits_{k\to\infty} x^k = \tilde{x}$, $\lim\limits_{k\to\infty} \varsigma^k = \tilde{\varsigma}$. 由引理 2.4 知 $\tilde{\varsigma} \in \partial c(\tilde{x})$.

引理 3.2 若点列 $\left\{x^k\right\}$ 收敛到 $\tilde{x}$, $\left\{y^k\right\}$ 收敛到 $\tilde{y}$, 则 $\lim\limits_{k\to\infty} P_{C_k}\left(y^k\right) = P_{C(\tilde{x})}(\tilde{y})$, 其中 $C(\tilde{x})$ 中 $\tilde{\varsigma}$ 满足 $\lim\limits_{k\to\infty} \varsigma^k = \tilde{\varsigma}$.

证 由引理 2.3 得

$$P_{C_k}\left(y^k\right)=y^k+\frac{\max\left\{0,\left(\varsigma^k\right)^T x^k-c\left(x^k\right)-\left(\varsigma^k\right)^T y^k\right\}}{\left\|\varsigma^k\right\|^2}\varsigma^k, \tag{3.2}$$

$$P_{C(\tilde{x})}\left(\tilde{y}\right)=\tilde{y}+\frac{\max\left\{0,\left(\tilde{\varsigma}\right)^T \tilde{x}-c\left(\tilde{x}\right)-\left(\tilde{\varsigma}\right)^T \tilde{y}\right\}}{\left\|\tilde{\varsigma}\right\|^2}\tilde{\varsigma}. \tag{3.3}$$

因为点列 $\left\{x^k\right\}$ 收敛到 $\tilde{x}$, $\left\{y^k\right\}$ 收敛到 $\tilde{y}$ 及 $\lim\limits_{k\to\infty}\varsigma^k=\tilde{\varsigma}$, 所以

$$\begin{aligned}\lim_{k\to\infty}P_{C_k}\left(y^k\right)&=\lim_{k\to\infty}y^k+\lim_{k\to\infty}\frac{\max\left\{0,\left(\varsigma^k\right)^T x^k-c\left(x^k\right)-\left(\varsigma^k\right)^T y^k\right\}}{\left\|\varsigma^k\right\|^2}\varsigma^k\\&=\tilde{y}+\frac{\max\left\{0,\left(\tilde{\varsigma}\right)^T \tilde{x}-c\left(\tilde{x}\right)-\left(\tilde{\varsigma}\right)^T \tilde{y}\right\}}{\left\|\tilde{\varsigma}\right\|^2}\tilde{\varsigma}\\&=P_{C(\tilde{x})}\left(\tilde{y}\right).\end{aligned}$$

证毕.

引理 3.3 设 x^* 为问题 (1.4) 的任意解, $\left\{x^k\right\}$ 为算法 3.2 产生的点列, 则数列 $\{\left\|x^k-x^*\right\|\}$ 收敛.

证 因为 x^* 为问题 (1.4) 的任意解, 所以 $x^*\in C\cap D$, $Ax^*\in Q$, 由引理 2.1(a) 得

$$\begin{aligned}\langle x^*-x^k, d_k\rangle&=\left\langle x^*-x^k, A^T\overline{d_k}\right\rangle\\&=\left\langle Ax^*-Ax^k, \overline{d_k}\right\rangle\\&=\langle A\,x^*+P_Q\left(Ax^k\right)-P_Q\left(Ax^k\right)-Ax^k, \overline{d_k}\rangle\\&=\left\|\overline{d_k}\right\|^2+\langle Ax^*-P_Q\left(Ax^k\right), P_Q\left(Ax^k\right)-Ax^k\rangle\\&\geq\left\|\overline{d_k}\right\|^2.\end{aligned}$$

又因为 $x^*\in C\subset C_k, x^*\in D\subset D_k$, 所以

$$\begin{aligned}\left\|x^{k+1}-x^*\right\|^2&=\left\|P_{D_k}P_{C_k}\left(x^k+w_kr_kd_k\right)-x^*\right\|^2\\&\leq\left\|P_{C_k}\left(x^k+w_kr_kd_k\right)-x^*\right\|^2\\&\leq\left\|x^k+w_kr_kd_k-x^*\right\|^2\\&=\left\|x^k-x^*\right\|^2-2w_kr_k\left\langle x^*-x^k, d_k\right\rangle+{w_k}^2{r_k}^2\|d_k\|^2\\&\leq\left\|x^k-x^*\right\|^2-2w_kr_k\left\|\overline{d_k}\right\|^2+{w_k}^2{r_k}^2\|d_k\|^2.\end{aligned} \tag{3.4}$$

由于 $0<\underline{w}\leq w_k\leq\overline{w}<2, k=0,1,2,\cdots$, 故

$$\left\|x^{k+1}-x^*\right\|^2\leq\left\|x^k-x^*\right\|^2-\underline{w}{r_k}^2\|d_k\|^2(2-\overline{w}). \tag{3.5}$$

从而数列 $\{\left\|x^k-x^*\right\|\}$ 收敛. 证毕.

注 7 由 (3.4) 式的最后一步可以得出, 当 $r_k=\dfrac{\|\overline{d_k}\|^2}{\|d_k\|^2}$ 时, $\|x^k-x^*\|$ 取得最小值, 也即 x^k 最接近 x^*.

定理 3.2 设 $\left\{x^k\right\}$ 为算法 3.2 产生的点列, 若问题 (1.4) 的解集非空, 则 $\left\{x^k\right\}$ 收敛到问题 (1.4) 的一个解.

证 设 x^* 为问题 (1.4) 的任意解, 由引理 3.2 知数列 $\{\left\|x^k-x^*\right\|\}$ 收敛, 同时序列 $\{x^k\},\{d_k\}$ 有界. 更进一步, 由 (3.5) 式得 $\lim\limits_{k\to\infty} {r_k}^2\|d_k\|^2=0$. 又因为 $\{d_k\}$ 有界, 所以 $\lim\limits_{k\to\infty} {r_k}^2\|d_k\|^4=0$, 即

$$\lim_{k\to\infty}\left\|\overline{d_k}\right\|^2=0.$$

也就是

$$\lim_{k\to\infty}\|P_{Q_k}\left(Ax^k\right)-Ax^k\|^2=0. \tag{3.6}$$

设 $\overline{x}$ 是 $\left\{x^k\right\}$ 的任意聚点, 则一定存在 $\left\{x^k\right\}$ 的子列 $\{x^{k_l}\}$, 使得 $\lim\limits_{k_l\to\infty} x^{k_l}=\overline{x}$. 下证 $\overline{x}$ 为问题 (1.4) 的一个解.

首先证明 $A\overline{x}\in Q$. 由 (3.6) 式得 $\lim\limits_{k_l\to\infty}\|P_{Q_{k_l}}\left(Ax^{k_l}\right)-Ax^{k_l}\|^2=0$. 显然 $P_{Q_{k_l}}(Ax^{k_l})\in Q_{k_l}$. 由 Q_{k_l} 的定义得

$$q(Ax^{k_l})+\langle\eta^{k_l},P_{Q_{k_l}}(Ax^{k_l})-Ax^{k_l}\rangle\le 0. \tag{3.7}$$

由引理 2.2 可知 $\{\eta^{k_l}\}$ 有界. 对 (3.7) 式取极限得 $q(A\overline{x})\le 0$. 即 $A\overline{x}\in Q$.

接下来证明 $\overline{x}\in C\cap D$.

由引理 2.1(b), (c) 可得

$$\begin{aligned}\left\|x^{k+1}-x^*\right\|^2&=\left\|P_{D_k}P_{C_k}(x^k+w_kr_kd_k)-x^*\right\|^2\\&\le\left\|P_{C_k}\left(x^k+w_kr_kd_k\right)-x^*\right\|^2-\left\|x^{k+1}-P_{C_k}\left(x^k+w_kr_kd_k\right)\right\|^2\\&\le\left\|x^k+w_kr_kd_k-x^*\right\|^2-\left\|x^{k+1}-P_{C_k}\left(x^k+w_kr_kd_k\right)\right\|^2.\end{aligned} \tag{3.8}$$

令 $\lim\limits_{k\to\infty}\left\|x^k-x^*\right\|^2=a\,(a\ge 0)$, 结合 $\lim\limits_{k\to\infty} {r_k}^2\|d_k\|^2=0$, 对 (3.8) 式取极限得

$$\lim_{k\to\infty}\left\|x^{k+1}-P_{C_k}\left(x^k+w_kr_kd_k\right)\right\|^2=0,$$

从而

$$\lim_{k_l\to\infty}\left\|x^{k_l+1}-P_{C_{k_l}}(x^{k_l}+w_{k_l}r_{k_l}d_{k_l})\right\|^2=0. \tag{3.9}$$

下证 $x^{k_l+1}\to\overline{x}\,(k_l\to\infty)$.

由于 $\{x^k\}$ 有界, 故 $\left\{x^{k_l+1}\right\}$ 有界. 设 $\overline{\overline{x}}$ 为 $\left\{x^{k_l+1}\right\}$ 的任意聚点, 易知 $\overline{\overline{x}}$ 也为 $\left\{x^k\right\}$ 的任意聚点, 则存在 $\left\{x^{k_l+1}\right\}$ 的子列 $\left\{x^{k_{l_i}+1}\right\}$, 使得 $\lim\limits_{k_{l_i}\to\infty} x^{k_{l_i}+1}=\overline{\overline{x}}$, 且 $\lim\limits_{k_{l_i}\to\infty} x^{k_{l_i}}=\overline{x}$.

由 (3.9) 式得

$$\lim_{k_{l_i}\to\infty}\left\|x^{k_{l_i}+1}-P_{C_{k_{l_i}}}(x^{k_{l_i}}+w_{k_{l_i}}r_{k_{l_i}}d_{k_{l_i}})\right\|^2=0. \tag{3.10}$$

由引理 3.2, 对 (3.10) 式取极限得

$$\bar{\bar{x}}=P_{C(\bar{x})}(\bar{x}), \tag{3.11}$$

其中 $C(\bar{x})=\{x\in R^N|c(\bar{x})+\langle\bar{\varsigma}\,,x-\bar{x}\rangle\le 0\},\bar{\varsigma}\in\partial c(\bar{x})$, 且 $\bar{\varsigma}$ 满足 $\lim\limits_{k_{l_i}\to\infty}\varsigma^{k_{l_i}}=\bar{\varsigma}$.

因为 $x^*\in C_{k_{li}}=\{x\in R^N|c\left(x^{k_{li}}\right)+\left\langle\varsigma^{k_{li}}\,,x-x^{k_{li}}\right\rangle\le 0\},\varsigma^{k_{li}}\in\partial c\left(x^{k_{li}}\right)$, 所以

$$c\left(x^{k_{li}}\right)+\left\langle\varsigma^{k_{li}}\,,x^*-x^{k_{li}}\right\rangle\le 0. \tag{3.12}$$

对 (3.12) 式取极限得 $c(\bar{x})+\langle\bar{\varsigma}\,,x^*-\bar{x}\rangle\le 0$, 从而 $x^*\in C(\bar{x})$. 由引理 2.1(c), 结合 (3.11) 式, 得

$$\left\|\overline{\overline{x}}-x^*\right\|^2\le\|\overline{x}-x^*\|^2-\left\|\overline{\overline{x}}-\overline{x}\right\|^2. \tag{3.13}$$

又因为 $\left\|\overline{\overline{x}}-x^*\right\|^2=\|\overline{x}-x^*\|^2$, 所以 $\overline{\overline{x}}=\overline{x}$. 从而有界子列 $\left\{x^{k_l+1}\right\}$ 有唯一聚点 $\overline{x}$.

由 (3.11) 式知 $\overline{x}=\overline{\overline{x}}=P_{C(\bar{x})}(\overline{x})$, 从而 $\overline{x}\in C(\bar{x})$. 由 $C(\bar{x})$ 的定义得

$$c(\bar{x})+\langle\bar{\varsigma}\,,\bar{x}-\bar{x}\rangle\le 0.$$

从而 $c(\overline{x})\le 0$, 即 $\overline{x}\in C$. 由于 D_k 为非空闭凸集, 且 $\{x^k\}\subset D^k$, 故 $\overline{x}\in D_k$, 再由 D_k 的构造得

$$f\left(x^{k_l}\right)+\left\langle\phi^{k_l}\,,\bar{x}-x^{k_l}\right\rangle\le 0. \tag{3.14}$$

由引理 2.2 知 $\{\phi^{k_l}\}$ 有界, 对 (3.14) 式取极限得 $f(\overline{x})\le 0$, 即 $\overline{x}\in D$.

综上所述, 可以得到 $\overline{x}$ 是问题 (1.4) 的一个解. 所以, 可以用 $\overline{x}$ 来代替上述证明过程中的 x^*. 又 $\{\left\|x^k-\overline{x}\right\|\}$ 收敛, 而其子列 $\{\left\|x^{k_l}-\overline{x}\right\|\}$ 收敛到零, 故 $\left\{x^k\right\}$ 收敛到 $\overline{x}$.

证毕.

4 数值实验

为了验证本文所给算法 3.2 的可行性, 给出如下几个例子. 下面的例 1–6 取 $s=1.2, s=3$ 两种情况来验证. 首先给出的例 1–4, 集合 C 与 Q 都是由可微凸函数的水平集来定义的; 其次给出的例 5–6, 集合 C 与 Q 为不可微凸函数的水平集来定义的. 最后给出的例 7 选取高维的水平集和随机的矩阵. 在实际计算中选取参数 $\omega_k=1$.

例 1 设 $C=\{x\in R^3|x_1^2-x_2\le 0\}$, $Q=\{x\in R^3|x_1^2+x_2^2+x_3^2\le 0\}$, $A=I$.

求 $x\in C, Ax\in Q$ s.t. $\|x\|_1\le s$.

例 2 设 $C=\{x\in R^3|x_1^2+x_2^2-1\le 0\}\,,Q=\{x\in R^3|x_2^2+x_3^2-1\le 0\}\,,A=\begin{bmatrix}2&1&0\\0&1&0\\0&0&2\end{bmatrix}$.

求 $x\in C, Ax\in Q$ s.t. $\|x\|_1\le s$.

例 3 设 $C=\{x\in R^3|x_2^2+x_3^2-4\le 0\}\,,Q=\{x\in R^3|x_3-x_1^2-1\le 0\}\,,A=I$.

求 $x\in C, Ax\in Q$ s.t. $\|x\|_1\le s$.

例 4 设 $C=\{x\in R^3|x_1+x_2^2+2x_3\le 0\}\,,Q=\{x\in R^3|x_1^3+x_2-x_3\le 0\}\,,A=\begin{bmatrix}1&1&0\\0&1&1\\2&0&1\end{bmatrix}$.

求 $x\in C, Ax\in Q$ s.t. $\|x\|_1\le s$.

例 5 设 $C=\{x\in R^3|x_1^2-|x_2|\le 0\}$, $Q=\{x\in R^3|x_1^2+x_2^2+x_3^2\le 0\}$, $A=I$.

求 $x\in C, Ax\in Q$ s.t. $\|x\|_1\le s$.

例 6 设 $C=\{x\in R^3|x_1+x_2^2+2|x_3|\le 0\}$, $Q=\{x\in R^3|x_1^3+|x_2|-x_3\le 0\}$, $A=\begin{bmatrix}1&1&0\\0&1&1\\2&0&1\end{bmatrix}$.

求 $x\in C, Ax\in Q$ s.t. $\|x\|_1\le s$.

例 7 设 $C=\{x\in R^N|\|x\|_2\le 0.25\}, Q=\{y\in R^M|0.6\le y_j\le 1, j=1,2\cdots,M\}, A=(a_{ij})_{M\times N}$ 且 $a_{ij}\in(0,50)$ 为随机选取的.

求 $x\in C, Ax\in Q$ s.t. $\|x\|_1\le s$.

例 1–6 的数值结果如下表 1.1–6.2. 在表 1.1–7.1 中, x^0 代表初始点, Iter 代表迭代步数, cpu 代表计算时间 (单位为秒), x^* 代表近似解 (以下这些结果都是运用 MATLAB 计算得到的).

表 1.1: 例 1 ($s=1.2$) 的数值结果

x^0	CPU	Iter	x^*
$(1,1,1)^T$	0.095	2	(0.4000, 0.4000, 0.4000)
$(1,2,3)^T$	0.091	3	(-0.0543, 0.3118, 0.8338)
$(10,10,10)^T$	0.093	2	(0.4000, 0.4000, 0.4000)
$(1,10,100)^T$	0.086	9	(-0.2168, 0.0685, 0.8695)

表 1.2: 例 1 ($s=3$) 的数值结果

x^0	CPU	Iter	x^*
$(1,1,1)^T$	0.14	5	(0.5773, 0.5773, 0.5773)
$(1,2,3)^T$	0.095	6	(0.2455, 0.5289, 0.8123)
$(10,10,10)^T$	0.084	6	(0.5773, 0.5773, 0.5773)
$(1,10,100)^T$	0.087	9	(-0.1327, 0.2424, 0.9610)

表 2.1: 例 2 ($s=1.2$) 的数值结果

x^0	CPU	Iter	x^*
$(1,1,1)^T$	0.090	2	(0.5323, 0.4147, 0.2529)
$(1,2,3)^T$	0.087	6	(0.0148, 0.4113, 0.4557)
$(10,10,10)^T$	0.134	5	(0.9999, 0.0030, -0.1676)
$(1,10,100)^T$	0.089	5	(-0.5335, 0.3244, 0.3420)

表 2.2: 例 2 ($s=3$) 的数值结果

x^0	CPU	Iter	x^*
$(1,1,1)^T$	0.086	5	(0.7822, 0.6227, 0.3912)
$(1,2,3)^T$	0.086	7	(0.5692, 0.7318, 0.3407)
$(10,10,10)^T$	0.095	6	(0.9634, 0.2677, 0.4029)
$(1,10,100)^T$	0.088	10	(0.5564, 0.7109, 0.3516)

表 3.1: 例 3 ($s = 1.2$) 的数值结果

x^0	CPU	Iter	x^*
$(1,2,3)^T$	0.081	2	(0.2333, 0.2333, 0.7333)
$(1,10,100)^T$	0.094	5	(-0.0769, -0.9647, 0.1583)
$(5,5,5)^T$	0.104	15	(1.2, 0, 0)

表 3.2: 例 3 ($s = 3$) 的数值结果

x^0	CPU	Iter	x^*
$(1,2,3)^T$	0.089	2	(0.8333, 0.8333, 1.3333)
$(1,10,100)^T$	0.087	7	(-0.2595, -1.9975, 0.0991)
$(5,5,5)^T$	0.089	2	(2.5333, 0.2333, 0.2333)

表 4.1: 例 4 ($s = 1.2$) 的数值结果

x^0	CPU	Iter	x^*
$(1,1,1)^T$	0.082	4	(0.2336, -0.0644, -0.1189)
$(-10,-10,-10)^T$	0.094	6	(0.0563, 0.0676, -1.0760)
$(10,10,10)^T$	0.106	14	(-0.0296, -0.0691, 0.0124)
$(1,10,100)^T$	0.109	19	(-0.0301, -0.0705, 0.0126)

表 4.2: 例 4 ($s = 3$) 的数值结果

x^0	CPU	Iter	x^*
$(1,\ 1,\ 1)^T$	0.086	4	(0.2336, -0.0644, -0.1189)
$(-10,-10,-10)^T$	0.085	6	(0.0725, 0.0202, -2.9072)
$(10,10,10)^T$	0.084	15	-0.0231, -0.0520, 0.0102)
$(1,10,100)^T$	0.087	20	(-0.0251, -0.0569, 0.0109)

表 5.1: 例 5 ($s = 1.2$) 的数值结果

x^0	CPU	Iter	x^*
$(1,\ 1,\ 1)^T$	0.045	2	(0.3216, 0.4392, 0.4392)
$(1,\ 2,\ 3)^T$	0.055	3	(-0.1691, 0.2588, 0.7720)
$(10,\ 10,\ 10)^T$	0.048	3	(-0.2178, -0.5879, 0.3943)
$(1,\ 10,\ 100)^T$	0.049	9	(0.0043, -0.1591, 0.9873)

表 5.2: 例 5 ($s = 3$) 的数值结果

x^0	CPU	Iter	x^*
$(1,\ 1,\ 1)^T$	0.045	5	(0.5504, 0.6455, 0.6455)
$(1,\ 2,\ 3)^T$	0.054	6	(0.2224, 0.5427, 0, 8338)
$(10,\ 10,\ 10)^T$	0.048	6	(-0.6430, 0.8498, 0.7373)
$(1,\ 10,\ 100)^T$	0.068	11	(0.2309, 0.0533, 0.9924)

表 6.1: 例 6 ($s = 1.2$) 的数值结果

x^0	CPU	Iter	x^*
$(1, 1, 1)^T$	0.051	183	(0.0000, -0.0000, 0.0000)
$(-10, -10, -10)^T$	0.061	413	(-0.5280, -0.6720, 0.0000)
$(10, 10, 10)^T$	0.067	264	(-0.5014, -0.6772, 0.0214)
$(1, 10, 100)^T$	0.068	267	(-0.5014, -0.6672, 0.0214)

表 6.2: 例 6 ($s = 3$) 的数值结果

x^0	CPU	Iter	x^*
$(1, 1, 1)^T$	0.057	183	(0.0000, -0.0000, 0.0000)
$(-10, -10, -10)^T$	0.048	5	(-1.9821, 0.0386, -0.0978)
$(10, 10, 10)^T$	0.047	7	(-0.9367, -0.3765, 0.3975)
$(1, 10, 100)^T$	0.052	10	(-1.8112, -0.1148, 0.8990)

例 7 的数值结果如下表, 由于选取的水平集为高维的、矩阵是随机的, 所以给出的计算结果中不显示近似解.

表 7.1: 例 7 ($s = 0.12$) 的数值结果

	M=50, N=55	M=100, N=100	M=150, N=200
x^0	CPU Iter	CPU Iter	CPU Iter
$(0, 0, \cdots ,0)^T$	0.045; 4	0.068; 5	0.056; 5
$100e_1$	0.058 ; 140	0.081; 172	0.126; 429
rand$(N, 1)$	0.075; 78	0.079; 97	0.067; 97
-100*rand$(N, 1)$	0.059; 151	0.069; 133	0.083; 143

从例 1–7 的数值实验结果可以看出, 所有的计算时间都在 0.1 秒左右完成, 甚至大多数的计算时间都在 0.1 秒以内. 由此可见松弛交替投影算法具有较强的可行性和实用性.

5 本文小结

本文主要对带 1 - 范数约束的分裂可行问题进行了求解. 首先, 在算法上取了可以直接计算的步长, 避免了在求合适的步长时进行大量的计算. 其次, 又寻找包含闭凸集的半空间, 用往半空间上的投影来代替往闭凸集上的投影, 从而提出松弛交替投影算法, 使得投影容易计算. 并给出几个实例来验证松弛交替投影算法的可行性. 数值实验表明本文的算法具有较强的可行性和实用性.

参 考 文 献

[1] Censor Y, Elfving T. A multiprojection algorithm using Bregman projection in a product space[J]. Numer. Alg., 1994, 8(2): 221–239.

[2] Liu B, Qu B. A successive projection scheme for solving the multiple-sets split feasibility problem[J]. Numer. Funct. Anal. Optim., 2014, 35(11): 1459–1466.

[3] Rockafellar R T. Convex analysis[M]. Vol. 28, Princeton: Princeton Math. Ser., 1970.

[4] 王宜举, 修乃华. 非线性最优化理论与方法 [M]. 北京: 北京科学出版社, 2012.

[5] Qu B, Xiu N. A note on the CQ algorithm for the split feasibility problem[J]. Inverse Prob., 2005, 21: 1655–1665.

[6] Censor Y, Elfving T, Kopf N, et al. The multiple-sets split feasibility problem and its applications for inverse problems[J]. Inverse Prob., 2005, 21: 2071–2084.

[7] Yang Q. The relaxed CQ algorithm solving the split feasibility problem [J]. Inverse Prob., 2004, 20(4): 1261–1266.

[8] Zhang H, Wang Y. A new CQ method for solving split feasibility problem [J]. Front. Math. China, 2005, 5(1): 37–46

[9] Facchinei F, Pang J S. Finite-dimensional variational inequalities and complementarity problems[M]. Vols. I and II, Berlin: Springer Verlag, 2003.

[10] Qu Biao, Liu Bing-hua and Zhengna. On the computation of the step-size for the CQ-like algorithms for the split feasibility problem[J]. Appl. Math. Comp., 2015, 262: 218–223.

[11] Carmi A, Censor Y and Gurfil P, Convex feasibility modeling and projection methods for sparse signal recovery[J]. J. Comp. Appl. Math., 2012, 236(17): 4318–4335.

[12] Amir Beck, Yonina C Eldar. Sparsity constrained nonlinear optimization conditions and algorithms[J]. SIAM Optim., 2012, 23(3): 1480–1509.

[13] Fukushima M. An outer approximation algorithm for solving general convex programs[J]. Opera. Res., 1983, 31: 101–113.

[14] He H, Ling C, Xu H. An implementable splitting algorithm for the 1-norm regularized split feasibility problem[J]. J. Sci. Comput., 2016, 67: 281–289.

[15] He S, Zhu W. A note on approximating curve with 1-norm regularization method for the split feasibilit problem[J]. J. Appl. Math., 2012, 16: 3800–3844.

[16] Massao Fukushima. 非线性最优化基础 [M]. 北京: 科学出版社, 2011.

[17] 兰晓坚, 曲彪. 求解分裂可行问题的一种半空间投影算法 [J]. 数学杂志, 2011, 31(3): 547–553.

[18] 房明磊, 朱志斌, 陈凤华, 张聪. 互补约束规划问题的一个广义梯度投影算法 [J]. 数学杂志, 2011, 31(4): 685–694.

PROJECTION ALGORITHMS FOR THE SOLUTION OF SPLIT FEASIBILITY PROBLEM WITH 1-NORM CONSTRAINTS

CHANG Han-xiao, QU Biao

(*School of Management, Qufu Normal University, Rizhao 276826, China*)

Abstract: In this paper, we mainly study the solution algorithm of the split feasibility problem subject to 1-norm constraints. By using the alternating projections algorithm, we solve the problem successfully and propose a relaxed alternating projections algorithm which improves the shortage of projecting directly onto the closed convex set. And we obtain the convergence of this algorithm.

Keywords: split feasible problem; 1-norm constraints; alternating projections; relaxed

2010 MR Subject Classification: 90C25; 65K05