凸优化和单调变分不等式的收缩算法

第七讲: 线性约束凸优化问题
基于松弛 PPA 的收缩算法

Relaxed PPA-based Contraction Methods for Linearly Constrained Convex Optimization

南京大学数学系　　何炳生
hebma@nju.edu.cn

The context of this lecture is based on the publications [6]

线性约束的凸优化问题

这一讲要求解的问题是

$$\min\{\theta(x) \mid Ax = b \, (or \;\; Ax \geq b) \;\; x \in \mathcal{X}\}$$

其中 $\theta(x)$ 是凸函数, $A \in \Re^{m\times n}$, $b \in \Re^m$, $\mathcal{X}$ 是 $\Re^n$ 中的闭凸集

对任意给定的 $r > 0$ 和 $a \in \Re^n$, 通篇我们假设子问题

$$\min\{\theta(x) + \frac{r}{2}\|x - a\|^2 \mid x \in \mathcal{X}\}$$

的求解是简单的.

目的 将线性约束的凸优化问题转换成单调变分不等式, 再用自调比方法, 生成类似于 PPA 的正则项. 我们并不要求正则项系数矩阵 Q 对称正定, 只要在迭代中都有 $(u^k - \tilde{u}^k)^T Q(u^k - \tilde{u}^k) \geq \tau\|u^k - \tilde{u}^k\|^2$. 基于松弛 PPA 的收缩算法是一类预测一校正方法, 产生预测点都用松弛 PPA 方法.

1 线性约束凸优化问题的 PPA 算法

1.1 线性约束凸优化问题与单调变分不等式

线性约束的凸优化问题

$$\min\{\theta(x) \mid Ax = b \,(or\;\, Ax \ge b)\; x \in \mathcal{X}\} \tag{1.1}$$

的 Lagrange 函数是定义在 $\mathcal{X} \times \Lambda$ 上的

$$L(x, \lambda) = \theta(x) - \lambda^T (Ax - b),$$

其中

$$\Lambda = \left\{ \begin{array}{lll} \Re^m, & \text{for the equality constraints} & Ax = b, \\ \Re^m_+, & \text{for the inequality constraints} & Ax \ge b. \end{array} \right.$$

设 (x^*, λ^*) 是 Lagrange 函数的一个鞍点, 便有

$$L_{\lambda\in\Lambda}(x^*, \lambda) \le L(x^*, \lambda^*) \le L_{x\in\mathcal{X}}(x, \lambda^*).$$

求 Lagrange 函数的一个鞍点等价于求 (x^*, λ^*) 使其满足:

$$\left\{\begin{array}{ll} x^* \in \mathcal{X}, & \theta(x)-\theta(x^*)+(x-x^*)^T(-A^T\lambda^*) \geq 0, \quad \forall\, x \in \mathcal{X}, \\ \lambda^* \in \Lambda, & (\lambda-\lambda^*)^T(Ax^*-b) \geq 0, \quad \forall\, \lambda \in \Lambda. \end{array}\right. \tag{1.2}$$

换句话说, 求 Lagrange 函数的鞍点等价于求解混合变分不等式

$$\text{VI}(\Omega, F, \theta) \quad u^* \in \Omega, \quad \theta(x)-\theta(x^*)+(u-u^*)^T F(u^*) \geq 0, \quad \forall\, u \in \Omega, \tag{1.3a}$$

的解, 其中

$$u = \begin{pmatrix} x \\ \lambda \end{pmatrix}, \quad F(u) = \begin{pmatrix} -A^T\lambda \\ Ax-b \end{pmatrix} \tag{1.3b}$$

和

$$\Omega = \mathcal{X} \times \Lambda. \tag{1.3c}$$

注意到 $\theta(x)$ 是凸函数, 此外 (1.3b) 中的算子 F 是仿射单调的.

我们通过求解相应的混合单调变分不等式来求解线性约束的凸优化问题.

1.2 H-模下的 PPA 算法和松弛 PPA 的收缩算法

邻近点算法 (PPA) 是求解混合单调变分不等式的一个经典方法. 用邻近点算法求解问题 (1.3), 对给定的 $u^k \in \Omega$ 和 $r > 0$, k-次迭代是求 $\tilde{u}^k \in \Omega$, 使得

$$\theta(x) - \theta(\tilde{x}^k) + (u - \tilde{u}^k)^T \{F(\tilde{u}^k) + r(\tilde{u}^k - u^k)\} \geq 0, \quad \forall u \in \Omega. \tag{1.4}$$

我们取 $\gamma \in [1, 2)$, 用公式 $u^{k+1} = u^k - \gamma(u^k - \tilde{u}^k)$ 产生下一个迭代点. 序列 $\{u^k\}$ 满足关系式

$$\|u^{k+1} - u^*\|^2 \leq \|u^k - u^*\|^2 - \gamma(2 - \gamma)\|u^k - \tilde{u}^k\|^2.$$

对解集 Ω^*, 算法产生的序列 $\{u^k\}$ 是 Fej$\acute{e}$r 单调的. 这些在前一讲已经有介绍.

直接求解 PPA 子问题 (1.4) 一般来说是办不到的. 为求解与凸优化问题 (1.1) 等价的变分不等式 (1.3), 上一讲提出的 Customized PPA 方法的子问题是

$$\tilde{u}^k \in \Omega, \quad \theta(x) - \theta(\tilde{x}^k) + (u - \tilde{u}^k)^T \{F(\tilde{u}^k) + H(\tilde{u}^k - u^k)\} \geq 0, \quad \forall u \in \Omega, \tag{1.5}$$

其中

$$H = \begin{pmatrix} rI_n & -A^T \\ -A & sI_m \end{pmatrix}.$$

注意到与讨论的优化问题等价的变分不等式 (1.2) 的具体结构 (见 (1.3b) 式), 子变分不等式 (1.5) 便是求 $(\tilde{x}^k, \tilde{\lambda}^k) \in \Omega$, 使得对一切 $(x, \lambda) \in \Omega$, 都有

$$\theta(x) - \theta(\tilde{x}^k) + \begin{pmatrix} x - \tilde{x}^k \\ \lambda - \tilde{\lambda}^k \end{pmatrix}^T \left\{ \begin{pmatrix} -A^T \tilde{\lambda}^k \\ A\tilde{x}^k - b \end{pmatrix} + \begin{pmatrix} r(\tilde{x}^k - x^k) - A^T(\tilde{\lambda}^k - \lambda^k) \\ -A(\tilde{x}^k - x^k) + s(\tilde{\lambda}^k - \lambda^k) \end{pmatrix} \right\} \geq 0, \quad \forall (x, \lambda) \in \Omega. \tag{1.6}$$

上式下半部分中未知的 $\tilde{x}^k$ 可以消掉, 问题 (1.6) 的一半松弛成

$$\tilde{\lambda}^k \in \Lambda, \quad (\lambda - \tilde{\lambda}^k)^T \{(Ax^k - b) + s(\tilde{\lambda}^k - \lambda^k)\} \geq 0, \quad \forall\, \lambda \in \Lambda.$$

这个 $\tilde{\lambda}^k$ 可以通过 $\tilde{\lambda}^k = P_\Lambda[\lambda^k - \frac{1}{s}(Ax^k - b)]$ 直接给出. 有了 $\tilde{\lambda}^k$, 通过求解极小化问题

$$\min \left\{ \theta(x) + \frac{r}{2} \left\| x - \left[x^k + \frac{1}{r} A^T(2\tilde{\lambda}^k - \lambda^k) \right] \right\|^2 \,\Big|\, x \in \mathcal{X} \right\} \tag{1.7}$$

得到的 $\tilde{x}^k$ 满足 (1.6) 的上半部分. 因为 (1.7) 中的 $\tilde{\lambda}^k$ 也是已知的, 极小化问题 (1.7) 的类型根据假设是容易求解的. 用公式 $u^{k+1} = u^k - \gamma(u^k - \tilde{u}^k)$ 产生

的迭代序列 $\{u^k\}$ 满足

$$\|u^{k+1} - u^*\|_H^2 \le \|u^k - u^*\|_H^2 - \gamma(2-\gamma)\|u^k - \tilde{u}^k\|_H^2.$$

说到底, 这个 H-模下的 PPA 算法, 也是对经典的 PPA 松弛后得到的收缩算法.

上述 H-模下 PPA 算法子问题 (1.6) 中的参数 r, s 要满足 $rs > \|A^TA\|$, 矩阵 H 才是正定的, 这才保证算法收敛. 有时估计 $\|A^TA\|$ 并不容易, 过大的估计相当于正则项太大, 会影响总体收敛速度. 这一讲试图解决这些问题.

1.3 松弛条件下 PPA 收缩算法

Idea of the Relaxed PPA 为求解变分不等式 (1.3), 在第 k-次迭代中, 经典的 PPA 是要求 $(\tilde{x}^k, \tilde{\lambda}^k) \in \Omega$ 使得

$$\theta(x) - \theta(\tilde{x}^k) + \begin{pmatrix} x - \tilde{x}^k \\ \lambda - \tilde{\lambda}^k \end{pmatrix}^T \left\{ \begin{pmatrix} -A^T\tilde{\lambda}^k \\ A\tilde{x}^k - b \end{pmatrix} + \begin{pmatrix} r(\tilde{x}^k - x^k) \\ s(\tilde{\lambda}^k - \lambda^k) \end{pmatrix} \right\} \ge 0, \ \ (x, \lambda) \in \Omega. \tag{1.8}$$

因为上面的子变分不等式的上下两部分都含有未知的 $\tilde{x}^k$ 和 $\tilde{\lambda}^k$, 直接求解一

般办不到. 如果将上半部分中的 $\tilde{\lambda}^k$ 松弛成 λ^k, 则变成求 $(\tilde{x}^k, \tilde{\lambda}^k) \in \Omega$ 使得

$$\theta(x) - \theta(\tilde{x}^k) + \begin{pmatrix} x - \tilde{x}^k \\ \lambda - \tilde{\lambda}^k \end{pmatrix}^T \left\{ \begin{pmatrix} -A^T \lambda^k \\ A\tilde{x}^k - b \end{pmatrix} + \begin{pmatrix} r(\tilde{x}^k - x^k) \\ s(\tilde{\lambda}^k - \lambda^k) \end{pmatrix} \right\} \ge 0, \ (x, \lambda) \in \Omega. \tag{1.9}$$

我们把子问题 (1.9) 称为松弛的 PPA (Relaxed PPA). 松弛以后的问题 (1.9) 是容易求解的. 首先 (1.9) 的上半 (Primal) 部分可以通过求解极小化问题

$$\min \left\{ \theta(x) + \frac{r}{2} \left\| x - \left[x^k + \frac{1}{r} A^T \lambda^k \right] \right\|^2 \,\Big|\, x \in \mathcal{X} \right\}$$

得到. 根据假设条件, 这是一个容易求解的问题. 有了 $\tilde{x}^k$, (1.9) 的下半 (Dual) 部分中要求的只剩下 $\tilde{\lambda}^k$, 它的数学形式是

$$\tilde{\lambda}^k \in \Lambda, \quad (\lambda - \tilde{\lambda}^k)^T \{ (A\tilde{x}^k - b) + s(\tilde{\lambda}^k - \lambda^k) \} \ge 0, \quad \forall \lambda \in \Lambda.$$

根据变分不等式和投影的关系, $\tilde{\lambda}^k$ 可以通过

$$\tilde{\lambda}^k = P_\Lambda [\lambda^k - \frac{1}{s}(A\tilde{x}^k - b)]$$

直接给出. 我们把 (1.9) 写成:

$$(\tilde{x}^k,\tilde{\lambda}^k)\in\Omega,\quad \theta(x)-\theta(\tilde{x}^k)+\begin{pmatrix} x-\tilde{x}^k\\ \lambda-\tilde{\lambda}^k\end{pmatrix}^T\left\{\begin{pmatrix} -A^T\tilde{\lambda}^k\\ A\tilde{x}^k-b\end{pmatrix}\right.$$
$$\left.+\begin{pmatrix} r(\tilde{x}^k-x^k)\\ 0\end{pmatrix}+\begin{pmatrix} A^T(\tilde{\lambda}^k-\lambda^k)\\ s(\tilde{\lambda}^k-\lambda^k)\end{pmatrix}\right\}\geq 0,\quad \forall(x,\lambda)\in\Omega.$$

它的紧凑形式是

$$\tilde{u}^k\in\Omega,\ \theta(x)-\theta(\tilde{x}^k)+(u-\tilde{u}^k)^T\{F(\tilde{u}^k)+Q(\tilde{u}^k-u^k)\}\geq 0,\ \forall u\in\Omega, \quad (1.10)$$

其中矩阵

$$Q=\begin{pmatrix} rI_n & A^T\\ 0 & sI_m\end{pmatrix}$$

是非对称的. 换句话说, 这样的检验点 (或预测点) $\tilde{u}^k=(\tilde{x}^k,\tilde{\lambda}^k)$ 是通过对经典的 PPA 子问题松弛求得的. 由于 (1.10) 中除了 $F(\tilde{u}^k)$ 外就是 $Q(\tilde{u}^k-u^k)$, 也可以把由此建立的收缩算法称为基于线性临近点项的 PPA 收缩算法. 收敛证明可见参考文献[6].

我们把由 (1.10) 中产生的点 $\tilde{u}^k$ 称为预测点. 将 (1.10) 中任意的 u 设为 u^*, 就能得到

$$(\tilde{u}^k - u^*)^T Q(u^k - \tilde{u}^k) \geq 0.$$

进而得到

$$(u^k - u^*)^T Q(u^k - \tilde{u}^k) \geq (u^k - \tilde{u}^k)^T Q(u^k - \tilde{u}^k).$$

只要 (1.10) 中的矩阵 Q 满足

$$Q^T + Q = \begin{pmatrix} 2rI_n & A^T \\ A & 2sI_m \end{pmatrix} \succ 0,$$

就可以构造收缩算法. 由上式可知, 当 $rs > \frac{1}{4}\|A^T A\|$ 时, $Q^T + Q$ 就正定. 这个条件显然比前一讲 H-模下的 PPA 算法对参数 r, s 要求 $rs > \|A^T A\|$ 来得宽松. 这是我们这一讲继续 Relaxed PPA 研究的理由.

在统一框架下的考虑 对给定的 u^k, 我们生成 $\tilde{u}^k \in \Omega$ 使得 (1.10) 成立, 其中 Q 是一个矩阵. 我们并不要求 Q 对称, 只要求存在一个 $\tau > 0$, 使得

$$(u^k - \tilde{u}^k)^T Q(u^k - \tilde{u}^k) \geq \tau\|u^k - \tilde{u}^k\|^2, \;\; \forall\, k \geq 0.$$

上述条件在 $Q^T + Q$ 正定时一定成立.

2 Primal-Dual 松弛 PPA 收缩算法

用松弛 PPA 按先 $\tilde{x}^k$ (primal) 后 $\tilde{\lambda}^k$ (dual) 的顺序生成预测点 $\tilde{u}^k = (\tilde{x}^k, \tilde{\lambda}^k)$, 然后构造的收缩算法, 称为基于Primal-Dual 松弛 PPA 的收缩算法 (Primal-dual relaxed PPA based contraction method).

2.1 Primal-Dual 生成预测点

Primal–Dual Method 生成预测点　对给定的 (x^k, λ^k) 和 $r > 0$, 通过求解

$$\min\Big\{\theta(x) + \frac{r}{2}\Big\|x - \big[x^k + \frac{1}{r}A^T\lambda^k\big]\Big\|^2 \,\Big|\, x \in \mathcal{X}\Big\} \tag{2.1a}$$

得到 Primal 预测点 $\tilde{x}^k$. 再选取适当的 $s > 0$ 并用

$$\tilde{\lambda}^k = P_\Lambda[\lambda^k - \frac{1}{s}(A\tilde{x}^k - b)] \tag{2.1b}$$

生成 Dual 预测点 $\tilde{\lambda}^k$ (如何选取 $s > 0$ 放在后面讨论).

注意到用 (2.1a) 生成的 Primal 预测点 $\tilde{x}^k$ 满足

$$\tilde{x}^k \in \mathcal{X}, \ \theta(x) - \theta(\tilde{x}^k) + (x - \tilde{x}^k)^T\{r(\tilde{x}^k - x^k) - A^T\lambda^k\} \ge 0, \ \forall x \in \mathcal{X}. \quad (2.2)$$

将 (2.1b) 改写成 $\tilde{\lambda}^k = P_\Lambda\{\tilde{\lambda}^k - [(\tilde{\lambda}^k - \lambda^k) + \frac{1}{s}(A\tilde{x}^k - b)]\}$, 则有

$$\tilde{\lambda}^k \in \Lambda, \quad (\lambda - \tilde{\lambda}^k)^T\{(A\tilde{x}^k - b) + s(\tilde{\lambda}^k - \lambda^k)\} \ge 0, \quad \forall \lambda \in \Lambda. \qquad (2.3)$$

把 (2.2) 和 (2.3) 写在一起, 有

$$(\tilde{x}^k, \tilde{\lambda}^k) \in \Omega, \quad \theta(x) - \theta(\tilde{x}^k) + \begin{pmatrix} x - \tilde{x}^k \\ \lambda - \tilde{\lambda}^k \end{pmatrix}^T \left\{ \begin{pmatrix} -A^T\tilde{\lambda}^k \\ A\tilde{x}^k - b \end{pmatrix} \right.$$

$$\left. + \begin{pmatrix} r(\tilde{x}^k - x^k) \\ 0 \end{pmatrix} + \begin{pmatrix} A^T(\tilde{\lambda}^k - \lambda^k) \\ s(\tilde{\lambda}^k - \lambda^k) \end{pmatrix} \right\} \ge 0, \quad \forall (x, \lambda) \in \Omega.$$

它的紧凑形式是

$$\tilde{u}^k \in \Omega, \ \theta(x) - \theta(\tilde{x}^k) + (u - \tilde{u}^k)^T\{F(\tilde{u}^k) + Q(\tilde{u}^k - u^k)\} \ge 0, \ \forall u \in \Omega, \quad (2.4)$$

其中矩阵 Q (对应于 Primal-Dual 方法生成预测点的 Q, 有时也记成 Q_{PD})

$$Q_{PD} = \begin{pmatrix} rI_n & A^T \\ 0 & sI_m \end{pmatrix}, \tag{2.5}$$

是非对称的.

如果 (2.4) 中的矩阵 Q 是对称正定的, 这个生成预测点 $\tilde{u}^k$ 的方法就是 PPA 算法. 由于这里的 Q 是非对称矩阵, 按照 [6] 中的说法, 这里的 Relaxed PPA 是带线性临近点项的 PPA.

以 u^* 替代 (2.4) 中的 $u \in \Omega$, 可得到

$$(\tilde{u}^k - u^*)^T Q(u^k - \tilde{u}^k) \geq \theta(\tilde{x}^k) - \theta(x^*) + (\tilde{u}^k - u^*)^T F(\tilde{u}^k). \tag{2.6}$$

因为 $\tilde{u}^k \in \Omega$ 并且 u^* 是 VI(Ω, F) 的解, 根据混合变分不等式 (1.3) 的定义, 有

$$\theta(\tilde{x}^k) - \theta(x^*) + (\tilde{u}^k - u^*)^T F(u^*) \geq 0.$$

由于 F 是单调算子, 因此有

$$(\tilde{u}^k - u^*)^T F(\tilde{u}^k) \geq (\tilde{u}^k - u^*)^T F(u^*).$$

根据上面二个关系式得到 (2.6) 的右端非负. 我们定义

$$\varphi(u^k, \tilde{u}^k) = (u^k - \tilde{u}^k)^T Q(u^k - \tilde{u}^k). \tag{2.7}$$

由 (2.6) 的右端非负得到下面的关键不等式

$$(u^k - u^*)^T Q(u^k - \tilde{u}^k) \geq \varphi(u^k, \tilde{u}^k), \quad \forall\, u^* \in \Omega^*. \tag{2.8}$$

这一节中的收缩算法, 都是基于向量 $(u^k - \tilde{u}^k)$ 的算法. 然而, 我们并不直接用 $(u^k - \tilde{u}^k)$, 而是用

$$d(u^k, \tilde{u}^k) = M(u^k - \tilde{u}^k) \tag{2.9}$$

作为寻查方向, 其中

$$M = D^{-1}Q, \tag{2.10}$$

矩阵 Q 由 (2.5) 给出, 矩阵

$$D = \begin{pmatrix} rI_n & 0 \\ 0 & sI_m \end{pmatrix}. \tag{2.11}$$

根据这些定义, 我们有

$$M = \begin{pmatrix} I_n & \frac{1}{r}A^T \\ 0 & I_m \end{pmatrix},$$

是对角部分为单位阵的上三角分块矩阵. 关系式 (2.7) 和 (2.8) 可以写成等价的

$$\varphi(u^k, \tilde{u}^k) = (u^k - \tilde{u}^k)^T DM(u^k - \tilde{u}^k). \tag{2.12}$$

和

$$\langle D(u^k - u^*), M(u^k - \tilde{u}^k)\rangle \geq \varphi(u^k, \tilde{u}^k), \quad \forall\, u^* \in \Omega^*. \tag{2.13}$$

2.2 初等的收缩算法

The Primary Contraction Methods (初等的收缩算法) 是指用确定的方向取单位步长的收缩算法.

The Primary Contraction Methods 对给定的 u^k 和由 (2.1) 生成的 $\tilde{u}^k$, 我们用

$$u^{k+1} = u^k - M(u^k - \tilde{u}^k) \tag{2.14}$$

生成新的迭代点. 在用 (2.1) 生成的 $\tilde{u}^k$ 的过程中, 采用 Armijo 法则, 可以做到

对给定的 $r > 0$ 和 $\nu \in (0,1)$, 选取 s 使得

$$\frac{1}{r}\|A^T(\lambda^k - \tilde{\lambda}^k)\|^2 \le \nu\big(r\|x^k - \tilde{x}^k\|^2 + s\|\lambda^k - \tilde{\lambda}^k\|^2\big). \tag{2.15}$$

特别当 s 取得使 $sr\nu \ge \|A^TA\|$ 时, 条件 (2.15) 自然成立.

我们考虑 D-模下的收缩算法. 根据迭代公式 (2.14) 有

$$\begin{aligned}
&\|u^k - u^*\|_D^2 - \|u^{k+1} - u^*\|_D^2 \\
&= \quad \|u^k - u^*\|_D^2 - \|(u^k - u^*) - M(u^k - \tilde{u}^k)\|_D^2 \\
&= \quad 2(u^k - u^*)^T DM(u^k - \tilde{u}^k) - \|M(u^k - \tilde{u}^k)\|_D^2.
\end{aligned}$$

对上式右端的 $(u^k - u^*)^T DM(u^k - \tilde{u}^k)$ 使用 (2.12)–(2.13), 我们有

$$\begin{aligned}
&\|u^k - u^*\|_D^2 - \|u^{k+1} - u^*\|_D^2 \\
&\ge \quad 2(u^k - \tilde{u}^k)^T DM(u^k - \tilde{u}^k) - \|M(u^k - \tilde{u}^k)\|_D^2.
\end{aligned} \tag{2.16}$$

注意到 (2.16) 的右端等于

$$
\begin{aligned}
& 2(u^k-\tilde{u}^k)^T DM(u^k-\tilde{u}^k) - \|M(u^k-\tilde{u}^k)\|_D^2 \\
= \quad & (u^k-\tilde{u}^k)^T \big[M^TD + DM - M^TDM\big](u^k-\tilde{u}^k).
\end{aligned} \tag{2.17}
$$

利用矩阵关系 $DM = Q$, 有

$$
M^TD + DM - M^TDM = Q^T + Q - Q^TM.
$$

经过简单计算就有

$$
\begin{aligned}
& M^TD + DM - M^TDM = Q^T + Q - Q^TM. \\
= \quad & \begin{pmatrix} 2rI_n & A^T \\ A & 2sI_m \end{pmatrix} - \begin{pmatrix} rI_n & 0 \\ A & sI_m \end{pmatrix} \begin{pmatrix} I_n & \frac{1}{r}A^T \\ 0 & I_m \end{pmatrix} \\
= \quad & \begin{pmatrix} rI_n & 0 \\ 0 & sI_m - \frac{1}{r}AA^T \end{pmatrix}.
\end{aligned}
$$

因此,

$$
\begin{aligned}
& 2(u^k-\tilde{u}^k)^T DM(u^k-\tilde{u}^k) - \|M(u^k-\tilde{u}^k)\|_D^2 \\
= \quad & \|u^k-\tilde{u}^k\|_D^2 - \frac{1}{r}\|A^T(\lambda^k-\tilde{\lambda}^k)\|^2.
\end{aligned} \tag{2.18}
$$

在条件 (2.15) 满足的情况下, 从 (2.18) 式得到

$$
\begin{aligned}
& 2(u^k-\tilde{u}^k)^T DM(u^k-\tilde{u}^k) - \|M(u^k-\tilde{u}^k)\|_D^2 \\
\geq \quad & (1-\nu)\big(r\|x^k-\tilde{x}^k\|^2 + s\|\lambda^k-\tilde{\lambda}^k\|^2\big).
\end{aligned} \tag{2.19}
$$

以 (2.19) 代入 (2.16), 就有下面的定理:

Theorem 2.1 在基于 *Primal-Dual* 松弛 *PPA* 的收缩算法中, 如果产生预测点时条件 (2.15) 成立, 则由初等收缩算法 (2.14) 生成的序列 $\{u^k=(x^k,\lambda^k)\}$ 满足

$$
\|u^{k+1}-u^*\|_D^2 \leq \|u^k-u^*\|_D^2 - (1-\nu)\|u^k-\tilde{u}^k\|_D^2. \tag{2.20}
$$

定理 2.1 是保证初等收缩算法收敛的关键不等式.

2.3 一般的收缩算法

一般收缩算法同样用给定的方向 $M(u^k - \tilde{u}^k)$, 但通过计算步长确定下一个迭代点. 在 D-模的意义下, 使新的迭代点靠解集尽可能近一些. 在用 (2.1) 生成的 $\tilde{u}^k$ 的过程中, 采用 Armijo 法则, 可以做到

对给定的 $r > 0$, 选取 s 使得

$$\begin{aligned}
&\frac{1}{s}\|A(x^k - \tilde{x}^k)\|^2 + \frac{1}{r}\|A^T(\lambda^k - \tilde{\lambda}^k)\|^2 \\
&\quad \le \quad 2\big(r\|x^k - \tilde{x}^k\|^2 + s\|\lambda^k - \tilde{\lambda}^k\|^2\big).
\end{aligned} \tag{2.21}$$

特别当 s 取得使 $sr \ge \frac{1}{2}\|A^TA\|$ 时, $\|A^TA\| \le 2rs$,

$$\begin{aligned}
&\frac{1}{s}\|A(x^k - \tilde{x}^k)\|^2 + \frac{1}{r}\|A^T(\lambda^k - \tilde{\lambda}^k)\|^2 \\
&\quad \le \quad \frac{1}{s}\|A^TA\| \cdot \|x^k - \tilde{x}^k\|^2 + \frac{1}{r}\|AA^T\| \cdot \|\lambda^k - \tilde{\lambda}^k\|^2,
\end{aligned}$$

条件 (2.21) 自然成立.

利用 D 和 M 的表达式以及 Cauchy-Schwarz 不等式, 我们有

$$\begin{aligned}
\varphi(u^k,\tilde{u}^k) &= (u^k-\tilde{u}^k)^T DM(u^k-\tilde{u}^k) \\
&= \|u^k-\tilde{u}^k\|_D^2 + (x^k-\tilde{x}^k)^T A^T(\lambda^k-\tilde{\lambda}^k) \\
&\geq r\|x^k-\tilde{x}^k\|^2 - \frac{1}{2}\|x^k-\tilde{x}^k\|\cdot\|A^T(\lambda^k-\tilde{\lambda}^k)\| \\
&\quad +s\|\lambda^k-\tilde{\lambda}^k\|^2 - \frac{1}{2}\|\lambda^k-\tilde{\lambda}^k\|\cdot\|A(x^k-\tilde{x}^k)\| \\
&\geq r\|x^k-\tilde{x}^k\|^2 - \frac{1}{4}\Big\{r\|x^k-\tilde{x}^k\|^2 + \frac{1}{r}\|A^T(\lambda^k-\tilde{\lambda}^k)\|^2\Big\} \\
&\quad +s\|\lambda^k-\tilde{\lambda}^k\|^2 - \frac{1}{4}\Big\{s\|\lambda^k-\tilde{\lambda}^k\|^2 + \frac{1}{s}\|A(x^k-\tilde{x}^k)\|^2\Big\} \\
&= \frac{3}{4}\Big\{r\|x^k-\tilde{x}^k\|^2 + s\|\lambda^k-\tilde{\lambda}^k\|^2\Big\} \\
&\quad -\frac{1}{4}\Big\{\frac{1}{s}\|A(x^k-\tilde{x}^k)\|^2 + \frac{1}{r}\|A^T(\lambda^k-\tilde{\lambda}^k)\|^2\Big\}.
\end{aligned}$$

在条件 (2.21) 满足的情况下, 根据上式就有

$$\varphi(u^k,\tilde{u}^k) = (u^k-\tilde{u}^k)^T DM(u^k-\tilde{u}^k) \geq \frac{1}{4}\|u^k-\tilde{u}^k\|_D^2. \tag{2.22}$$

$\boxed{\textbf{The General Contraction Method I}}$ 对给定的 u^k 和由 (2.1) 生成的 $\tilde{u}^k$, 用

$$u(\alpha) = u^k - \alpha M(u^k - \tilde{u}^k) \tag{2.23}$$

产生依赖于步长 α 的迭代点. 对任意给定的 $u^* \in \Omega^*$, 我们将

$$\vartheta(\alpha) = \|u^k - u^*\|_D^2 - \|u(\alpha) - u^*\|_D^2 \tag{2.24}$$

看成是本次迭代的 “进步量” , 它是步长 α 的函数. 利用 (2.23) 和 (2.24) 中 $\vartheta(\alpha)$ 的定义, 我们有

$$\begin{aligned}\vartheta(\alpha) &= \|u^k - u^*\|_D^2 - \|(u^k - u^*) - \alpha M(u^k - \tilde{u}^k)\|_D^2 \\ &= 2\alpha(u^k - u^*)^T DM(u^k - \tilde{u}^k) - \alpha^2\|M(u^k - \tilde{u}^k)\|_D^2.\end{aligned}$$

对上式右端的 $(u^k - u^*)^T DM(u^k - \tilde{u}^k)$ 使用 (2.12) 和 (2.13), 就有

$$\vartheta(\alpha) \ge q(\alpha),$$

其中

$$q(\alpha) = 2\alpha(u^k - \tilde{u}^k)^T DM(u^k - \tilde{u}^k) - \alpha^2\|M(u^k - \tilde{u}^k)\|_D^2. \tag{2.25}$$

同样, 注意到 (2.25) 中的 $q(\alpha)$ 是 α 的二次函数, 它在

$$\alpha_k^* = \frac{(u^k - \tilde{u}^k)^T DM(u^k - \tilde{u}^k)}{\|M(u^k - \tilde{u}^k)\|_D^2} = \frac{(u^k - \tilde{u}^k)^T Q(u^k - \tilde{u}^k)}{(M(u^k - \tilde{u}^k))^T Q(u^k - \tilde{u}^k)} \tag{2.26}$$

时取得极大值.

当条件 (2.21) 满足时, 对所有的 $k \geq 0$, 都有 $\alpha^* \geq 1/6$.

根据 (2.26), 为证明上述结论, 只要证明

$$6(u^k - \tilde{u}^k)^T DM(u^k - \tilde{u}^k) - \|M(u^k - \tilde{u}^k)\|_D^2 \geq 0. \tag{2.27}$$

由 (2.22), 我们有

$$\begin{aligned} &6(u^k - \tilde{u}^k)^T DM(u^k - \tilde{u}^k) - \|M(u^k - \tilde{u}^k)\|_D^2 \\ \geq\ & \|u^k - \tilde{u}^k\|_D^2 + 2(u^k - \tilde{u}^k)^T DM(u^k - \tilde{u}^k) - \|M(u^k - \tilde{u}^k)\|_D^2. \end{aligned}$$

再利用 (2.18), 得到

$$\begin{aligned} &\|u^k - \tilde{u}^k\|_D^2 + 2(u^k - \tilde{u}^k)^T DM(u^k - \tilde{u}^k) - \|M(u^k - \tilde{u}^k)\|_D^2 \\ =\ & 2\|u^k - \tilde{u}^k\|_D^2 - \frac{1}{r}\|A^T(\lambda^k - \tilde{\lambda}^k)\|^2. \end{aligned}$$

当条件 (2.21) 满足时上式右端非负, (2.27) 成立, 就有 $\alpha_k^* \geq 1/6$.

我们想要极大化 $\vartheta(\alpha)$ (见 (2.24)), 由于它含有未知 u^*, 我们不得已才极大化它的下界函数 $q(\alpha)$. 在实际计算中, 我们取

$$u^{k+1} = u^k - \gamma\alpha_k^* M(u^k - \tilde{u}^k), \tag{2.28}$$

为新的迭代点, 其中 $\gamma \in [1,2)$ 称为松弛因子.

利用 $\vartheta(\alpha) \geq q(\alpha)$, 将 (2.25) 中的 α 置换成 $\gamma\alpha^*$, 并用 (2.26), 就有

$$\begin{aligned}\|u^{k+1} - u^*\|_D^2 &\leq \|u^k - u^*\|_D^2 - q(\gamma\alpha^*) \\ &= \quad \|u^k - u^*\|_D^2 - \gamma(2-\gamma)\alpha_k^*(u^k - \tilde{u}^k)^T DM(u^k - \tilde{u}^k).\end{aligned}$$

根据上面的不等式, 由 $\alpha_k^* \geq 1/6$ 和关系式 (2.22), 就得到下面的定理

Theorem 2.2 在基于*Primal-Dual* 松弛 *PPA* 的收缩算法中*,* 如果产生预测点时条件 (2.21) 成立*,* 则由一般收缩算法 (2.28) 生成的序列$\{u^k = (x^k, \lambda^k)\}$ 满足

$$\|u^{k+1} - u^*\|_D^2 \leq \|u^k - u^*\|_D^2 - \frac{\gamma(2-\gamma)}{24}\|u^k - \tilde{u}^k\|_D^2. \tag{2.29}$$

迭代序列 $\{u^k\}$ 是 D-模下 Fejér 单调的.

The General Contraction Method II 对给定的 u^k 和由 (2.1) 生成的 $\tilde{u}^k$, 我们也可以用

$$u^{k+1} = u^k - \gamma\alpha_k^* Q^{-T} D(u^k - \tilde{u}^k) \tag{2.30}$$

生成新的迭代点, 由于矩阵

$$Q^{-T}D = \begin{pmatrix} \frac{1}{r}I_n & 0 \\ -\frac{1}{rs}A & \frac{1}{s}I_m \end{pmatrix}\begin{pmatrix} rI_n & 0 \\ 0 & sI_m \end{pmatrix} = \begin{pmatrix} I_n & 0 \\ -\frac{1}{s}A & I_m \end{pmatrix}$$

的形式是简单的, 校正 (2.30) 容易实现, 步长则由

$$\alpha_k^* = (u^k - \tilde{u}^k)^T Q(u^k - \tilde{u}^k)/\|u^k - \tilde{u}^k\|_D^2, \qquad \gamma \in (0,2)$$

给出. 由 (2.22), 可知 $\alpha_k^* \geq 1/4$. 对 $H = QD^{-1}Q^T$, 利用 (2.12) 和 (2.13), 则有

$$\begin{aligned} \|u^{k+1} - u^*\|_H^2 &= \|u^k - u^* - \gamma\alpha^* Q^{-T}D(u^k - \tilde{u}^k)\|_H^2 \\ &\leq \|u^k - u^*\|_H^2 - 2\gamma\alpha_k^*(u^k - \tilde{u}^k)^T Q(u^k - \tilde{u}^k) + \gamma^2(\alpha_k^*)^2\|u^k - \tilde{u}^k\|_D^2 \\ &= \|u^k - u^*\|_H^2 - \gamma(2-\gamma)\alpha_k^*(u^k - \tilde{u}^k)^T Q(u^k - \tilde{u}^k). \\ &\leq \|u^k - u^*\|_H^2 - \frac{\gamma(2-\gamma)}{16}\|u^k - \tilde{u}^k\|_D^2. \end{aligned}$$

由于 u^* 是任意给定的解点, 这样生成的序列 $\{u^k\}$ 在 H-模下 Fejér 单调的.

3 Dual-Primal 松弛 PPA 收缩算法

通过对经典PPA (1.8) 中下半部分中的 $\tilde{x}^k$ 松弛成 x^k 构造 Relaxed PPA. 子问题 (1.8) 就简化成求 $(\tilde{x}^k, \tilde{\lambda}^k) \in \Omega$, 使得对任意的 $u = (x, \lambda) \in \Omega$, 都有

$$\theta(x) - \theta(\tilde{x}^k) + \begin{pmatrix} x - \tilde{x}^k \\ \lambda - \tilde{\lambda}^k \end{pmatrix}^T \left\{ \begin{pmatrix} -A^T\tilde{\lambda}^k \\ Ax^k - b \end{pmatrix} + \begin{pmatrix} r(\tilde{x}^k - x^k) \\ s(\tilde{\lambda}^k - \lambda^k) \end{pmatrix} \right\} \geq 0. \tag{3.1}$$

松弛以后的子问题 (3.1) 是容易求解的. 首先对 (3.1) 的下半 (Dual) 部分, 根据变分不等式和投影的关系, $\tilde{\lambda}^k$ 可以通过

$$\tilde{\lambda}^k = P_\Lambda[\lambda^k - \frac{1}{s}(Ax^k - b)]$$

直接给出. 有了 $\tilde{\lambda}^k$, (3.1) 的上半 (Primal) 部分中要求的只剩下 $\tilde{x}^k$, 它可以通过求解极小化问题

$$\min\left\{\theta(x) + \frac{r}{2}\big\|x - \big[x^k + \frac{1}{r}A^T\tilde{\lambda}^k\big]\big\|^2 \,\big|\, x \in \mathcal{X}\right\}$$

得到. 根据假设条件, 这是一个容易求解的问题.

用松弛 PPA 按先 $\tilde{\lambda}^k$ (dual) 后 $\tilde{x}^k$ (primal) 的顺序生成预测点 $\tilde{u}^k = (\tilde{x}^k, \tilde{\lambda}^k)$, 然后构造的收缩算法, 我们称之为基于Dual-Primal 松弛 PPA 的收缩算法 (Dual - primal relaxed PPA based contraction method).

3.1 Dual-Primal 生成预测点

Dual–Primal Method 生成预测点 对给定的 (x^k, λ^k) 和 $s > 0$, 由

$$\tilde{\lambda}^k = P_\Lambda\{\lambda^k - \frac{1}{s}(Ax^k - b)\} \tag{3.2a}$$

生成 Dual 预测点 $\tilde{\lambda}^k$. 然后选取适当的 $r > 0$ 并求解

$$\min\Big\{\theta(x) + \frac{r}{2}\big\|x - \big[x^k + \frac{1}{r}A^T\tilde{\lambda}^k\big]\big\|^2 \,\Big|\, x \in \mathcal{X}\Big\} \tag{3.2b}$$

得到 Primal 预测点 $\tilde{x}^k$ (如何选取 $r > 0$ 放在后面讨论).

根据前面的分析 (见 (3.1)), 由 (3.2) 生成的预测点 $(\tilde{x}^k, \tilde{\lambda}^k)$ 满足

$$(\tilde{x}^k, \tilde{\lambda}^k) \in \Omega, \quad \theta(x) - \theta(\tilde{x}^k) + \begin{pmatrix} x - \tilde{x}^k \\ \lambda - \tilde{\lambda}^k \end{pmatrix}^T \left\{ \begin{pmatrix} -A^T\tilde{\lambda}^k \\ A\tilde{x}^k - b \end{pmatrix} \right.$$
$$\left. + \begin{pmatrix} r(\tilde{x}^k - x^k) \\ -A(\tilde{x}^k - x^k) \end{pmatrix} + \begin{pmatrix} 0 \\ s(\tilde{\lambda}^k - \lambda^k) \end{pmatrix} \right\} \geq 0, \quad \forall (x, \lambda) \in \Omega.$$

它的紧凑形式是

$$\tilde{u}^k \in \Omega,\ \theta(x) - \theta(\tilde{x}^k) + (u - \tilde{u}^k)^T\{F(\tilde{u}^k) + Q(\tilde{u}^k - u^k)\} \geq 0,\ \forall u \in \Omega, \tag{3.3}$$

其中矩阵(对应于 Dual-Primal 方法生成预测点的 Q, 有时也记成 Q_{DP})

$$Q_{DP} = \begin{pmatrix} rI_n & 0 \\ -A & sI_m \end{pmatrix}, \tag{3.4}$$

是非对称的. 类似地, 可以得到下面的关键不等式

$$(\tilde{u}^k - u^*)^T Q(u^k - \tilde{u}^k) \geq 0, \quad \forall\ u^* \in \Omega^*. \tag{3.5}$$

采用

$$M(u^k - \tilde{u}^k), \qquad (\text{其中} \quad M = D^{-1}Q) \tag{3.6}$$

作为寻查方向, 其中矩阵 H 如 (2.14) 给出. 根据这些定义, 我们有

$$M(u^k - \tilde{u}^k) = \begin{pmatrix} x^k - \tilde{x}^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ -\frac{1}{s}A & 0 \end{pmatrix} \begin{pmatrix} x^k - \tilde{x}^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix}. \tag{3.7}$$

同时有

$$(u^k - u^*)^T DM(u^k - \tilde{u}^k) \geq \varphi(u^k, \tilde{u}^k), \quad \forall\, u^* \in \Omega^*, \tag{3.8}$$

其中

$$\varphi(u^k, \tilde{u}^k) = (u^k - \tilde{u}^k)^T DM(u^k - \tilde{u}^k). \tag{3.9}$$

3.2 初等的收缩算法

这里的收缩算法是一种预测校正方法. 初等的收缩算法 (Primary Contraction Methods) 是在选取确定的方向后, 校正产生新迭代点时取单位步长的算法.

The Primary Contraction Methods

对给定的 u^k 和由 (3.2) 生成的 $\tilde{u}^k$, 我们用

$$u^{k+1} = u^k - M(u^k - \tilde{u}^k) \tag{3.10}$$

生成新的迭代点. 在用 (3.2) 生成的 $\tilde{u}^k$ 的过程中, 采用 Armijo 法则, 可以做到

对给定的$s > 0$ 和 $\nu \in (0,1)$, 选取 r 使得

$$\frac{1}{s}\|A(x^k - \tilde{x}^k)\|^2 \le \nu\big(r\|x^k - \tilde{x}^k\|^2 + s\|\lambda^k - \tilde{\lambda}^k\|^2\big). \tag{3.11}$$

特别当 s 取得使 $sr\nu \ge \|A^TA\|$ 时, 条件 (3.11) 自然成立.

与定理 2.1 类似, 对基于Dual-Primal 松弛 PPA 的收缩算法, 有如下的定理:

Theorem 3.1 在基于*Dual-Primal* 松弛 *PPA* 的收缩算法中, 如果产生预测点时条件 (3.11) 成立, 则由初等收缩算法 (3.10) 生成的序列 $\{u^k = (x^k, \lambda^k)\}$ 满足

$$\|u^{k+1} - u^*\|_D^2 \le \|u^k - u^*\|_D^2 - (1-\nu)\|u^k - \tilde{u}^k\|_D^2. \tag{3.12}$$

定理 3.1 是保证初等收缩算法收敛的关键不等式.

3.3 一般的收缩算法

一般收缩算法用同样的给定方向 $M(u^k - \tilde{u}^k)$, 通过计算步长确定下一个迭代点. 在 H-模的意义下, 使新的迭代点靠解集尽可能近一些.

对给定的 $s > 0$, 选取 r 使得

$$\begin{aligned} &\frac{1}{s}\|A(x^k - \tilde{x}^k)\|^2 + \frac{1}{r}\|A^T(\lambda^k - \tilde{\lambda}^k)\|^2 \\ \leq\quad & 2\big(r\|x^k - \tilde{x}^k\|^2 + s\|\lambda^k - \tilde{\lambda}^k\|^2\big). \end{aligned} \tag{3.13}$$

这是能够办到的. 当 s 取得使 $sr \geq \frac{1}{2}\|A^TA\|$ 时, 条件 (3.13) 自然成立.

类似与 (2.21)–(2.22) 的分析, 在条件 (3.13) 满足的情况下, 有

$$\varphi(u^k, \tilde{u}^k) = (u^k - \tilde{u}^k)^T DM(u^k - \tilde{u}^k) \geq \frac{1}{4}\|u^k - \tilde{u}^k\|_D^2. \tag{3.14}$$

一般的收缩算法 I　对给定的 u^k 和由 (3.2) 生成的 $\tilde{u}^k$, 我们取

$$u^{k+1} = u^k - \gamma\alpha_k^* M(u^k - \tilde{u}^k), \tag{3.15}$$

为新的迭代点, 其中

$$\alpha_k^* = (u^k - \tilde{u}^k)^T DM(u^k - \tilde{u}^k)/\|M(u^k - \tilde{u}^k)\|_D^2 \tag{3.16}$$

$\gamma \in [1, 2)$ 称为松弛因子. 同样, 在条件 (3.13) 满足时, 会有 $\alpha_k^* > 1/6$.

Theorem 3.2 在基于*Dual-Primal* 松弛 *PPA* 的收缩算法中, 如果产生预测点时条件 (3.13) 成立, 则由收缩算法 (3.15) 生成的序列 $\{u^k = (x^k, \lambda^k)\}$ 满足

$$\|u^{k+1} - u^*\|_D^2 \le \|u^k - u^*\|_D^2 - \frac{\gamma(2-\gamma)}{24}\|u^k - \tilde{u}^k\|_D^2. \tag{3.17}$$

同样, 迭代序列 $\{u^k\}$ 是 D-模下 Fejér 单调的, 上式是证明收敛的关键不等式.

一般的收缩算法 II

同样, 对给定的 u^k 和由 (3.2) 生成的 $\tilde{u}^k$, 也可以用

$$u^{k+1} = u^k - \gamma\alpha_k^* Q^{-T} D(u^k - \tilde{u}^k) \tag{3.18}$$

生成新的迭代点, 其中 $\alpha_k^* = \varphi(u^k, \tilde{u}^k)/\|u^k - \tilde{u}^k\|_D^2$, 取 $H = QD^{-1}Q^T$, 就有

$$\|u^{k+1} - u^*\|_H^2 \le \|u^k - u^*\|_H^2 - \frac{\gamma(2-\gamma)}{16}\|u^k - \tilde{u}^k\|_D^2.$$

由于 u^* 是任意给定的解点, 这样生成的序列 $\{u^k\}$ 在 H-模下 Fejér 单调.

4 数值计算

我们以第四讲中试验的相关性矩阵校正和矩阵完整化问题作为例子, 并与前一讲的 PPA 算法比较计算效果. 我们采用基于Dual-Primal 松弛 PPA 的收缩算法.

4.1 相关性矩阵校正问题

相关性矩阵校正问题的数学形式是

$$\min\{\frac{1}{2}\|X-C\|_F^2 \mid \mathrm{diag}(X)=e,\ X\in S_+^n\}, \tag{4.1}$$

在第四讲的 §4.1 中已经作了介绍. 用 $z\in\Re^n$ 作为等式约束 $\mathrm{diag}(X)=e$ 的 Lagrange 乘子. 我们用 (3.2) 生成问题 (4.1) 的预测点, 问题 (3.2b) 是求解

$$\min\{\frac{1}{2}\|X-C\|_F^2+\frac{r}{2}\|X-[X^k+\frac{1}{r}\mathrm{diag}(\tilde{z}^k)]\|_F^2|X\in S_+^n\}. \tag{4.2}$$

这样的工作在前一讲的 §4.1 作了介绍, 求 $(\tilde{x}^k,\tilde{\lambda}^k)$ 的主要工作是做 EIG 分解.

Relaxed PPA-Primal method 只是将前一讲 Code 4.1 (Classical PPA) 的第 (9) 行的 `A=(X0*r + C + diag(yt*2 -y0))/(1+r)` 改成 `A=(X0*r + C + diag(yt))/(1+r)` 并将 (12) 行的 `y = y0 - EY` 改成 `y = y0 - (EY - diag(EX)/s)`.

Code 4.1. Relaxed PPA – Primal method $rs = 1.01, \ s = 0.5$

```
%%%    RePPA (primal mothed) for calibrating correlation matrix   %(1)
function RePPA_MP(n,C,r,s,tol);                                   %(2)
X=eye(n);      y=zeros(n,1);    tic;      %%  The initial iterate %(3)
stopc=1;                         k=0;                             %(4)
while (stopc>tol && k<=100)         %% Beginning of an Iteration  %(5)
 if mod(k,20)==0 fprintf(’ k=%4d   epsm=%9.3e  \n’,k,stopc); end; %(6)
    X0=X;     y0=y;     k=k+1;                                    %(7)
    yt=y0 - (diag(X0)-ones(n,1))/s;              EY=y0-yt;        %(8)
    A=(X0*r + C + diag(yt))/(1+r);                                %(9)
    [V,D]=eig(A);   D=max(0,D);  XT=(V*D)*V’;     EX=X0-XT;      %(10)
    ex=max(max(abs(EX)));  ey=max(abs(EY));   stopc=max(ex,ey);  %(11)
    X=X0 - EX;             y=y0 - (EY - diag(EX)/s;              %(12)
end;                                         % End of an Iteration %(13)
toc;                  TB = max(abs(diag(X-eye(n))));             %(14)
fprintf(’ k=%4d  epsm=%9.3e   max|X_jj - 1|=%8.5f \n’,k,stopc,TB); %%
```

采用 Relaxed-PPA-Prmal method 与 Classical-PPA method 计算效果完全一样.

我们下面比较 Relaxed PPA – general Method 与前一讲的 Extended PPA Method 的计算效率.

从 Relaxed PPA – Primal Method 到 Relaxed PPA – general Method, 其差别在:

- Code 4.2 与 Code 4.1 的前 11 行完全相同.
- 在 Code 4.2 中, 增加的 12–14 行是为了计算步长.
- Code 4.2 中的第 15 行给出新的迭代点, 与 Code 4.1 中给出新迭代点的第 11 行相比, 寻查方向相同, 只是步长不同. Code 4.1 中用的是单位步长. Code 4.2 中的步长要用公式 (3.16) 计算.
- 为使用 Relaxed PPA-Primal Method, 需要满足条件 $rs \geq \|A^T A\|$ (见 (3.11)), 由于 $\|A^T A\| = 1$, 我们取 $rs = 1.01$, $s = 0.5$.
- 对使用 Relaxed PPA-General Method, 只需要满足条件 $rs \geq \frac{1}{2}\|A^T A\|$ (见 (3.13)), 我们取 $rs = 0.65$, $s = 0.4$. 相对于步长为 1 的算法, 这里生成预测点时, 将参数 s 和 r 都乘上一个因子 0.8.

Code 4.2 Relaxed PPA – general method $rs = 0.65,\ s = 0.4$

```
%%%  RePPA_MG (general mothed)for calibrating correlation matrix  %(1)
function RePPA_MG(n,C,r,s,tol,gamma)                               %(2)
X=eye(n);      y=zeros(n,1);    tic;      %%  The initial iterate  %(3)
stopc=1;                     k=0;                                  %(4)
while (stopc>tol && k<=100)         %% Beginning of an Iteration  %(5)
 if mod(k,20)==0 fprintf(’ k=%4d   epsm=%9.3e  \n’,k,stopc); end; %(6)
    X0=X;    y0=y;    k=k+1;                                       %(7)
    yt=y0 - (diag(X0)-ones(n,1))/s;               EY=y0-yt;        %(8)
    A=(X0*r + C + diag(yt))/(1+r);                                 %(9)
    [V,D]=eig(A);   D=max(0,D);  XT=(V*D)*V’;     EX=X0-XT;       %(10)
    ex=max(max(abs(EX)));  ey=max(abs(EY));   stopc=max(ex,ey);  %(11)
    T1=EX(:)’*EX(:);     T2=EY(:)’*EY(:);                         %(12)
    dEX=diag(EX);        T12 =EY’*dEX;        T3=dEX’*dEX;        %(13)
    alpha=(T1*r + T2*s - T12)/(T1*r + T2*s - T12*2 + T3/s);       %(14)
    X=X0-EX*(alpha*gamma);    y=y0-(EY-dEX/s)*(alpha*gamma);      %(15)
end;                                      % End of an Iteration %(16)
toc;                    TB = max(abs(diag(X-eye(n))));            %(17)
fprintf(’ k=%4d  epsm=%9.3e   max|X_jj - 1|=%8.5f \n’,k,stopc,TB); %%
```

矩阵校正问题 (4.1)–使用 Matlab 中的 eig 子程序

$n \times n$ Matrix	Extended PPA		Relaxed PPA	
$n =$	No. It	CPU Sec.	No. It	CPU Sec.
100	22	0.24	22	0.24
200	25	1.42	22	1.24
500	27	11.66	22	9.62
800	29	50.47	23	39.77
1000	31	99.26	25	78.95
2000	41	883.76	33	713.36

♣ 关于相关系数矩阵校正的程序在附件的 Codes-05 的文件夹“矩阵校正”的 MaT-EIG 中. 只要运行 demo.m, 输入 n 就可以了.
其中的 PPAC.m 和 PPAG.m 分别是 Classical PPA 和 Extended PPA 的子程序.
RePPAMP.m 和 RePPAMG.m 分别是 Relaxed PPA-Primal Method 和 Relaxed PPA-General Method 的子程序.

♣ 如果改用 Kim TOH 写的 mexeig 做 $[V,D]=$ mexeig(A), 计算时间大为节省. 相应的程序在附件的 Codes-05 的文件夹"矩阵校正" 的 Mex-EIG 中.

矩阵校正问题 (4.1)–特征值分解使用 mexeig

$n \times n$ Matrix	Extended PPA		Relaxed PPA	
$n=$	No. It	CPU Sec.	No. It	CPU Sec.
100	22	0.09	22	0.09
200	25	0.37	22	0.32
500	27	3.35	22	2.67
800	29	11.56	23	9.12
1000	31	22.58	25	18.20
2000	39	209.90	33	168.79

从计算实践看, 对同样的问题, Relaxed PPA-General Method 的计算效果比 Extended PPA 要好. 这两个方法生成预测点的工作量是相同的, 虽然用 Relaxed PPA-General Method 需要计算步长, 因为计算步长是 $O(n^2)$ 的运算, Relaxed PPA-General Method 花费的时间还是有较大幅度的节省.

4.2 矩阵完整化方面的应用

矩阵完整化问题的试验例子与前一讲的 §4.2 相同, 算例来自 [1]. 我们只将 Relaxed PPA-General Method 与 Extended PPA 做比较. 这两个方法生成预测点的工作量是相同的.

♣ 矩阵完整化用 Matlab SVD 的程序在附件的 Codes-05 的文件夹 矩阵完整化–SVD-MaT 中. 只要运行 demo.m 就可以了. 要对不同情形试验, 只要在 demo.m 中用 % 做适当选择. 其中的 PPAGMaT.m 和 PPAMMaT.m 分别是 Extended PPA 和 Relaxed PPA 的子程序, 均采用 $\gamma = 1.5$.

如果用 Matlab 中的标准 SVD, 对同样规模的问题, 计算时间与迭代次数成正比. 虽然用 Relaxed PPA-General Method 需要计算步长, 因为计算步长是 $O(n^2)$ 的运算, 这在总的计算量中是微不足道的. 我们将Relaxed PPA-General Method 求解矩阵完整化问题的程序作为 Code 4.3 列在后面.

用 Relaxed PPA-General method 求解矩阵完整化问题与前一讲的 Code 4.3 (Extended PPA) 的差别是：将第 (10) 行的

`A=X0 + (YT*2 -Y0)/r` 改成 `A = X0 + (YT)/r` .

在 Code 4.3 (Extended PPA) 第 (13) 行以后加上三行计算步长 $\alpha = \gamma\alpha^*$.

并将 Code 4.3 (Extended PPA) 的 (15)-(16) 行改成了现在的 (18)-(19) 行.

矩阵完整化问题：用 Matlab 中标准SVD 求解结果

Unknown $n \times n$ matrix M				Extended PPA		Relaxed PPA	
n	rank(ra)	m/d_{ra}	m/n^2	#iters	times(Sec.)	#iters	times(Sec.)
1000	10	6	0.12	77	867.82	56	629.67
1000	50	4	0.39	37	411.28	27	302.27
1000	100	3	0.58	31	362.58	29	305.93

因此, 用 Relaxed PPA-General Method 花费的时间有较大幅度的节省.

♣ 注意到, [1] 中的方法对这三个例子的迭代次数分别是 117, 114 和 129 (See the first three examples in Table 5.1 of [1], pp. 1974). 他们的方法每次迭代的主要工作量也是做一次 SVD 分解, 由于采用了不完全分解技术, [1] 中节省了每次 SVD 分解的运行时间. 我们调用的是 Matlab 中标准SVD, 减少了迭代次数, 但没有节省总的运行时间.

Code 4.3. Relaxed PPA for Matrix Completion Problem

```
function PPAE(n,r,s,M,Omega,maxIt,tol,gamma)     %  Itital Process %%(1)
X=zeros(n);      Y=zeros(n);       YT=zeros(n);                     %(2)
  nM0=norm(M(Omega),'fro');    eps=1;  VioKKT=1;  k=0;   tic;       %(3)
%% Minimum nuclear norm solution by PPA method                      %(4)
while  (eps > tol && k<= maxIt)                                     %(5)
 if mod(k,5)==0                                                     %(6)
 fprintf('It=%3d |X-M|/|M|=%9.2e VioKKT=%9.2e\n',k,eps,VioKKT); end;%(7)
  k=k+1;       X0=X;      Y0=Y;                                     %(8)
  YT(Omega)=Y0(Omega)-(X0(Omega)-M(Omega))/s;             EY=Y-YT;  %(9)
  A = X0 + (YT)/r;                          [U,D,V]=svd(A,0);      %(10)
  D=D-eye(n)/r;     D=max(D,0);      XT=(U*D)*V';          EX=X-XT;  %(11)
  DXM=XT(Omega)-M(Omega);           eps = norm(DXM,'fro')/nM0;     %(12)
  VioKKT = max( max(max(abs(EX)))*r, max(max(abs(EY))) );          %(13)
  T1=EX(:)'*EX(:);       T2=EY(:)'*EY(:);                           %(14)
  T12 = EY(:)'*EX(:);    EXOm = EX(Omega);  T3 = EXOm(:)'*EXOm(:);  %(15)
  alpha =(T1*r + T2*s - T12)*gamma/(T1*r + T2*s - T12*2 + T3/s);   %(16)
   if (eps <= tol)       alpha=1; end;                              %(17)
  X = X0 - EX*alpha;                                                %(18)
  Y(Omega) = Y0(Omega) -  (EY(Omega) - EXOm/s)*alpha;               %(19)
end                                                                 %(20)
 fprintf('It=%3d  |X-M|/|M|=%9.2e  Vi0KKT=%9.2e \n',k,eps,VioKKT); %(21)
   RelEr=norm((X-M),'fro')/norm(M,'fro');   toc;                    %(22)
 fprintf(' Relative error = %9.2e  Rank(X)=%3d \n',RelEr,rank(X)); %(23)
 fprintf(' Violation of KKT Condition = %9.2e  \n',VioKKT);        %(24)
```

4.3 Min-Max 问题上的应用

对第一讲 §4.2 中提到用全变差极小处理图像去模糊 [2] 的 min-max 问题,

$$\min_{x\in\mathcal{X}} \max_{y\in\mathcal{Y}} \;\Phi(x,y) := \theta_1(x) - y^T Ax - \theta_2(y). \tag{4.3}$$

它可以转换成等价的变分不等式: 求 $u^* = (x^*, y^*) \in \mathcal{X}\times\mathcal{Y}$, 使得

$$\theta(u) - \theta(u^*) + \begin{pmatrix} x - x^* \\ y - y^* \end{pmatrix}^T \begin{pmatrix} -A^T y^* \\ Ax^* \end{pmatrix} \ge 0, \quad \forall (x,y) \in \mathcal{X}\times\mathcal{Y}, \tag{4.4}$$

其中 $\theta(u) = \theta_1(x) + \theta_2(y)$. 用这一讲介绍的 Relaxed PPA 方法去求解, 对给定的 (x^k, y^k), 可以通过

$$\boxed{\begin{aligned} \tilde{x}^k &= \text{Argmin}\{\theta_1(x) - x^T A^T y^k + \frac{r}{2}\|x - x^k\|^2 \mid x \in \mathcal{X}\}. && \text{(4.5a)} \\ \tilde{y}^k &= \text{Argmin}\Big\{\theta_2(y) + y^T A\tilde{x}^k + \frac{s}{2}\|y - y^k\|^2 \,\Big|\, y \in \mathcal{Y}\Big\}. && \text{(4.5b)} \end{aligned}}$$

或者

$$\tilde{y}^k = \text{Argmin}\{\theta_2(y) + y^T A x^k + \frac{s}{2}\|y - y^k\|^2 \,|\, y \in \mathcal{Y}\}. \tag{4.6a}$$

$$\tilde{x}^k = \text{Argmin}\{\theta_1(x) - x^T A^T \tilde{y}^k + \frac{r}{2}\|x - x^k\|^2 \,\big|\, x \in \mathcal{X}\}. \tag{4.6b}$$

由 (4.5) 生成的 $(\tilde{x}^k, \tilde{y}^k) \in \mathcal{X} \times \mathcal{Y}$, 对一切 求 $(x, y) \in \mathcal{X} \times \mathcal{Y}$, 都有

$$\theta(u) - \theta(\tilde{u}^k) + \begin{pmatrix} x - \tilde{x}^k \\ y - \tilde{y}^k \end{pmatrix}^T \left\{ \begin{pmatrix} -A^T y^k \\ A\tilde{x}^k \end{pmatrix} + \begin{pmatrix} r(\tilde{x}^k - x^k) \\ s(\tilde{y}^k - y^k) \end{pmatrix} \right\} \geq 0. \tag{4.7}$$

由 (4.6) 生成的 $(\tilde{x}^k, \tilde{y}^k) \in \mathcal{X} \times \mathcal{Y}$, 对一切 求 $(x, y) \in \mathcal{X} \times \mathcal{Y}$, 都有

$$\theta(u) - \theta(\tilde{u}^k) + \begin{pmatrix} x - \tilde{x}^k \\ y - \tilde{y}^k \end{pmatrix}^T \left\{ \begin{pmatrix} -A^T \tilde{y}^k \\ Ax^k \end{pmatrix} + \begin{pmatrix} r(\tilde{x}^k - x^k) \\ s(\tilde{y}^k - y^k) \end{pmatrix} \right\} \geq 0. \tag{4.8}$$

无论是 (4.7) 还是 (4.8), 都可以写成

$$\tilde{u}^k \in \Omega, \qquad (u - \tilde{u}^k)^T \{F(\tilde{u}^k) + Q(\tilde{u}^k - u^k)\} \geq 0, \quad \forall\, u \in \Omega, \tag{4.9}$$

的形式, 所不同的只是

$$Q=\begin{pmatrix} rI_n & A^T \\ 0 & sI_m \end{pmatrix} \qquad \text{和} \qquad Q=\begin{pmatrix} rI_n & 0 \\ -A & sI_m \end{pmatrix}.$$

都是当 $rs>\frac{1}{4}\|A^TA\|$ 时, 矩阵 Q^T+Q 正定. 我们建议取 $rs\approx 0.7\|A^TA\|$.

对由 (2.11) 定义的矩阵 D, 新的迭代点u^{k+1} 用迭代式

$$u^{k+1}=u^k-\gamma\alpha_k^*M(u^k-\tilde{u}^k), \quad \gamma\in[1,2),$$

生成(一般取 $\gamma=1.5$). 采用

$$M=D^{-1}Q, \qquad \alpha_k^*=\frac{(u^k-\tilde{u}^k)^TQ(u^k-\tilde{u}^k)}{\|M(u^k-\tilde{u}^k)\|_D^2}, \tag{4.10}$$

生成的序列 $\{u^k\}$ 在 D-模下 Fejér 单调; 若采用

$$M=Q^{-T}D, \qquad \alpha_k^*=\frac{(u^k-\tilde{u}^k)^TQ(u^k-\tilde{u}^k)}{\|u^k-\tilde{u}^k\|_D^2}, \tag{4.11}$$

生成的序列 $\{u^k\}$ 则在 H-模下 Fejér 单调的($H=QD^{-1}Q^T$). 类似的结论可参阅本讲的 §2.3 和 §3.3. 用这类方法求解图像去模糊问题可以参考 [8].

References

[1] J. F. Cai, E. J. Candès and Z. W. Shen, A singular value thresholding algorithm for matrix completion, SIAM J. Optim., **20**, 1956-1982, 2010.

[2] A. Chambolle and T. Pock, A first-order primal-dual algorithms for convex problem with applications to imaging, J. Math. Imaging Vison, **40**, 120-145, 2011.

[3] B.S. He, A new method for a class of linear variational inequalities, Math. Progr., **66**, 137-144, 1994.

[4] B.S. He, A class of projection and contraction methods for monotone variational inequalities, Applied Mathematics and Optimization, **35**, 69-76, 1997.

[5] B.S. He, Inexact implicit methods for monotone general variational inequalities, Math. Program. Series A, **86**, 199-217, 1999

[6] B. S. He, X. L. Fu and Z.K. Jiang, Proximal point algorithm using a linear proximal term, JOTA **141**, 209-239, 2009.

[7] B. S. He and M.-H. Xu, A general framework of contraction methods for monotone variational inequalities, Pacific J. Optimization, **4**, 195-212, 2008.

[8] B.S. He and X. M. Yuan, Convergence analysis of primal-dual algorithms for a saddle-point problem: From contraction perspective, SIAM Journal on Imaging Science, **5**, 119-149, 2012.

[9] R. M. Larsen, Propack − software for large and sparse svd calculation, http://sun.stanford.edu/rmunk/PROPACK/, 2005.

[10] B. Martinet, Regularisation, d'inéquations variationelles par approximations succesives, Rev. Francaise d'Inform. Recherche Oper., **4**, 154-159, 1970.

[11] R.T. Rockafellar, Monotone operators and the proximal point algorithm, SIAM J. Control Optim., **14**, 877-898, 1976.