生活理念对设计优化分裂算法的帮助
— 以改造 ADMM 求解三个可分离算子问题为例

何 炳 生

南京大学数学系　　南京大学管理科学与工程国际研究中心

Email: hebma@nju.edu.cn　Homepage: math.nju.edu.cn/~hebma

摘要. 乘子交替方向法(Alternating Directions Method of Multipliers), 简称交替方向法 (ADMM), 是求解含两个可分离算子的凸优化问题的有效工具, 人们很想将这类算法推广到多个算子的问题上. 对三个算子的问题, 直接推广的方法试算很多例子照样收敛, 但生活理念的直觉告诉我们, 由于存在某种不公平性, 在一般条件下要保证理论收敛不太可能. 在致力于给出一些能够求解多个可分离算子凸优化问题的修正 ADMM 方法后, 举出了直接推广的 ADMM 对三个或者三个以上算子问题不收敛的例子. 本文从生活理念的角度, 解释为什么直接推广的 ADMM 对三个算子的问题不能保证收敛. 将 ADMM 改造成能求解含三个可分离算子的问题, 又是根据的什么生活理念. 数学知识与生活理念相辅相成, 是我职业生涯的一个主要体验.

关键词. 可分离算子的凸优化, 交替方向法, 分裂收缩算法.

1 引 言

交替方向法是求解凸优化的一类分裂收缩算法. 这类方法的迭代过程中, 要求解的子问题往往是一个简单的凸优化. 它的目标函数是两个凸函数的和, 其中一个凸函数不一定光滑. 说这类子问题简单, 是指问题的解有显式表达式, 或是我们不太费劲就能求得一个符合要求的近似解. 通篇, 在分析算法收敛性时, 我们常常要用到下面的引理.

Lemma 1.1. 设 $\mathcal{X} \subset \Re^n$ 是闭凸集, $\theta(x)$ 和 $f(x)$ 都是凸函数, 其中 $f(x)$ 可微. 记 x^* 是凸优化问题 $\min\{\theta(x)+f(x)\,|\,x\in\mathcal{X}\}$ 的解. 我们有

$$x^* = \arg\min\{\theta(x)+f(x)\,|\,x\in\mathcal{X}\} \tag{1.1a}$$

的充分必要条件是

$$x^*\in\mathcal{X}, \quad \theta(x)-\theta(x^*)+(x-x^*)^T\nabla f(x^*)\ge 0, \quad \forall\, x\in\mathcal{X}. \tag{1.1b}$$

通俗的说, 如果用极大代替极小, 这个引理就相当于瞎子爬山原理.

2 线性约束的凸优化

由于本文讨论的方法都和增广 Lagrange 乘子法有关, 我们先从线性约束的凸优化谈起. 设 $\mathcal{X}\subset\Re^n$ 是闭凸集, $A\in\Re^{m\times n}$, $b\in\Re^m$. 考虑线性约束的凸优化问题

$$\min\{\theta(x)\mid Ax=b,\ x\in\mathcal{X}\}. \tag{2.1}$$

它的 Lagrange 函数是定义在 $\mathcal{X}\times\Re^m$ 上的

$$L(x,\lambda)=\theta(x)-\lambda^T(Ax-b). \tag{2.2}$$

2.1 Lagrange 函数 (2.2) 鞍点等价的变分不等式

我们讨论问题 (2.1) 的 Lagrange 函数 (2.2) 的鞍点. 如果一对 $(x^*,\lambda^*)\in\mathcal{X}\times\Re^m$ 满足

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

则称为 Lagrange 函数 (2.2) 的鞍点. 上面的不等式关系可以写成

$$\left\{\begin{array}{ll} x^*\in\mathcal{X}, & L(x,\lambda^*)-L(x^*,\lambda^*)\ge 0,\quad \forall\, x\in\mathcal{X},\\ \lambda^*\in\Re^m, & L(x^*,\lambda^*)-L(x^*,\lambda)\ge 0,\quad \forall\,\lambda\in\Re^m.\end{array}\right.$$

也就是

$$\left\{\begin{array}{lcl} x^* &=& \arg\min\{L(x,\lambda^*)|x\in\mathcal{X}\},\\ \lambda^* &=& \arg\max\{L(x^*,\lambda)|\lambda\in\Re^m\}.\end{array}\right.$$

利用 (2.2) 和引理 1.1, 上述优化问题的最优性条件是

$$\left\{\begin{array}{lr} x^*\in\mathcal{X}, & \theta(x)-\theta(x^*)+(x-x^*)^T(-A^T\lambda^*)\ge 0,\quad \forall\, x\in\mathcal{X},\\ \lambda^*\in\Re^m, & (\lambda-\lambda^*)^T(Ax^*-b)\ge 0,\quad \forall\,\lambda\in\Re^m.\end{array}\right.$$

换句话说, 鞍点就是变分不等式

$$w^*\in\Omega,\quad \theta(x)-\theta(x^*)+(w-w^*)^TF(w^*)\ge 0,\quad \forall\, w\in\Omega, \tag{2.3a}$$

的解, 其中

$$w=\begin{pmatrix} x\\ \lambda\end{pmatrix},\quad F(w)=\begin{pmatrix} -A^T\lambda\\ Ax-b\end{pmatrix}\quad 和\quad \Omega=\mathcal{X}\times\Re^m. \tag{2.3b}$$

注意到, 这样的 F, 总满足 $(w-\bar{w})^T(w-F(\bar{w}))=0$.

2.2 求解问题 (2.1) 的增广 Lagrange 乘子法

凸优化问题 (2.1) 的的增广 Lagrange 函数由它的 Lagrange 函数 $L(x,\lambda)$ (2.2) 和一个由等式线性约束的二次函数 $\frac{\beta}{2}\|Ax-b\|^2$ 组成, 即

$$\mathcal{L}_\beta(x,\lambda)=\theta(x)-\lambda^T(Ax-b)+\frac{\beta}{2}\|Ax-b\|^2,\qquad (\beta>0\ \text{ 是等式约束的罚参数}).$$

增广 Lagrange 乘子法(Augmented Lagrangian Method)

增广 Lagrange 乘子法 (ALM) 的 k-次迭代从一个给定的 λ^k 开始, 通过

$$\left\{\begin{array}{lcl} x^{k+1} &=& \arg\min\big\{\mathcal{L}_\beta(x,\lambda^k)\;\big|\; x\in\mathcal{X}\big\},\\ \lambda^{k+1} &=& \lambda^k-\beta(Ax^{k+1}-b),\end{array}\right. \tag{2.4}$$

完成, 为下一次迭代提供了一个新的 λ^{k+1}. 在变分不等式 (2.3) 中, x 是原始变量, λ 是对偶变量. 而在算法 (2.4) 中, x^{k+1} 是根据 λ^k 计算得来的结果. 因此, 我们称

x 为算法的中间变量 (intermidiate variable), λ 为核心变量 (essential variable).

根据优化问题的最优性条件 (1.1), 算法 (2.4) 提供的 $w^{k+1}=(x^{k+1},\lambda^{k+1})\in\Omega$ 满足

$$\left\{\begin{array}{rcl} \theta(x)-\theta(x^{k+1})\;+\;(x-x^{k+1})^T\{-A^T\lambda^k+\beta A^T(Ax^{k+1}-b)\} &\ge& 0,\ \ \forall x\in\mathcal{X},\\ (\lambda-\lambda^{k+1})^T\{(Ax^{k+1}-b)+\frac{1}{\beta}(\lambda^{k+1}-\lambda^k)\} &\ge& 0,\ \ \forall\lambda\in\Re^m.\end{array}\right.$$

进一步利用关系式 $\lambda^{k+1} = \lambda^k - \beta(Ax^{k+1} - b)$, 上式可以改写成:

$$w^{k+1} \in \Omega, \;\; \theta(x) - \theta(x^{k+1}) + \begin{pmatrix} x - x^{k+1} \\ \lambda - \lambda^{k+1} \end{pmatrix}^T \begin{pmatrix} -A^T\lambda^{k+1} \\ (Ax^{k+1} - b) + \frac{1}{\beta}(\lambda^{k+1} - \lambda^k) \end{pmatrix} \ge 0, \;\; \forall\, w \in \Omega.$$

接着利用 (2.3) 的记号, 可以把上式写成更紧凑的式子:

$$\begin{aligned} w^{k+1} \in \Omega, \qquad & \theta(x) - \theta(x^{k+1}) + (w - x^{k+1})^T F(w^{k+1}) \\ & \ge (\lambda - \lambda^{k+1})^T \frac{1}{\beta}(\lambda^k - \lambda^{k+1}), \quad \forall w \in \Omega. \end{aligned} \tag{2.5}$$

变分不等式 (2.5) 中, 如果 $\lambda^k = \lambda^{k+1}$, 那么 w^{k+1} 就是变分不等式 (2.3) 的解. 将 (2.5) 中的任意 $w \in \Omega$ 设成一个固定的解点 w^*, 我们得到

$$(\lambda^{k+1} - \lambda^*)^T \frac{1}{\beta}(\lambda^k - \lambda^{k+1}) \ge \theta(x^{k+1}) - \theta(x^*) + (w^{k+1} - w^*)^T F(w^{k+1}). \tag{2.6}$$

利用 $(w^{k+1} - w^*)^T F(w^{k+1}) = (w^{k+1} - w^*)^T F(w^*)$ 和 w^* 的最优性, 即

$$\theta(x^{k+1}) - \theta(x^*) + (w^{k+1} - w^*)^T F(w^*) \ge 0,$$

推得 (2.6) 的右端非负, 最终得到

$$(\lambda^{k+1} - \lambda^*)^T (\lambda^k - \lambda^{k+1}) \ge 0. \tag{2.7}$$

在恒等式

$$\|b\|^2 = \|a\|^2 - \|a - b\|^2 - 2b^T(a - b),$$

中置 $a = \lambda^k - \lambda^*$ 和 $b = \lambda^{k+1} - \lambda^*$, 并利用 (2.7), 就有

$$\|\lambda^{k+1} - \lambda^*\|^2 \le \|\lambda^k - \lambda^*\|^2 - \|\lambda^k - \lambda^{k+1}\|^2. \tag{2.8}$$

这个核心变量序列 $\{\lambda^k\}$ 收缩的不等式, 是增广 Lagrange 乘子法收敛的关键保证.

3 两个可分离算子的凸优化问题

考虑两个可分离算子的线性约束凸优化问题

$$\min\{\theta_1(x) + \theta_2(y) \mid Ax + By = b,\, x \in \mathcal{X}, y \in \mathcal{Y}\}. \tag{3.1}$$

它的 Lagrange 函数是

$$L^2(x, y, \lambda) = \theta_1(x) + \theta_2(y) - \lambda^T(Ax + By - b). \tag{3.2}$$

3.1 Lagrange 函数 (3.2) 鞍点等价的变分不等式

与 §2.1 中同样的分析告知我们, Lagrange 函数 (3.2) 的鞍点 $\big((x^*, y^*), \lambda^*\big)$ 是变分不等式

$$w^* \in \Omega, \quad \theta(u) - \theta(u^*) + (w - w^*)^T F(w^*) \ge 0, \quad \forall\, w \in \Omega, \tag{3.3a}$$

的解, 其中

$$w = \begin{pmatrix} x \\ y \\ \lambda \end{pmatrix}, \quad u = \begin{pmatrix} x \\ y \end{pmatrix}, \qquad F(w) = \begin{pmatrix} -A^T\lambda \\ -B^T\lambda \\ Ax + By - b \end{pmatrix}, \tag{3.3b}$$

$$\theta(u) = \theta_1(x) + \theta_2(y), \qquad \text{和} \qquad \Omega = \mathcal{X} \times \mathcal{Y} \times \Re^m.$$

凸优化问题 (3.1) 的的增广 Lagrange 函数由它的 Lagrange 函数 $L^2(x,y,\lambda)$ (3.2) 和等式线性约束的二次函数 $\frac{\beta}{2}\|Ax+By-b\|^2$ 组成, 即

$$\mathcal{L}^2_\beta(x,y,\lambda)=\theta_1(x)+\theta_2(y)-\lambda^T(Ax+By-b)+\frac{\beta}{2}\|Ax+By-b\|^2, \tag{3.4}$$

其中 $\beta>0$ 是等式约束的罚参数. 如果我们用增广 Lagrange 乘子法求解问题 (3.1), k-次迭代从一个给定的 λ^k 开始, 通过

$$\left\{\begin{array}{rcl}(x^{k+1},y^{k+1}) &=& \arg\min\big\{\mathcal{L}^2_\beta(x,y,\lambda^k)\;\big|\;x\in\mathcal{X},y\in\mathcal{Y}\big\},\\ \lambda^{k+1} &=& \lambda^k-\beta(Ax^{k+1}+By^{k+1}-b),\end{array}\right. \tag{3.5}$$

完成. 此时, 迭代中的核心变量还是 λ，还会有 (2.8) 这样的收缩不等式. **方法的缺点**是问题 (3.1) 的可分离性质没有得到利用！

3.2 求解问题 (3.1) 的乘子交替方向法

用乘子交替方向法(Alternating Directions Method of Multipliers) [4, 5] 求解问题 (3.1), k-次迭代从给定的 (y^k,λ^k) 开始, 通过

$$\left\{\begin{array}{l}x^{k+1}=\arg\min\{\mathcal{L}^2_\beta(x,y^k,\lambda^k)\,|\,x\in\mathcal{X}\},\\ y^{k+1}=\arg\min\{\mathcal{L}^2_\beta(x^{k+1},y,\lambda^k)\,|\,y\in\mathcal{Y}\},\\ \lambda^{k+1}=\lambda^k-\beta(Ax^{k+1}+By^{k+1}-b),\end{array}\right. \tag{3.6}$$

完成, 为下一次迭代提供了一个新的 (y^{k+1},λ^{k+1}).

- 在乘子交替方向法 (3.6) 中, x-子问题和 y-子问题是利用了问题 (3.1) 的可分离性质分别求解的.
- 从 (3.5) 到 (3.6), 乘子交替方向法(ADMM)可以看作是松弛了的增广 Lagrange 乘子法(ALM).

ADMM 方法 (3.6) 求解 (3.1), x^{k+1} 是由给定的 (y^k,λ^k), 通过计算得到的, 因此是中间变量. 我们把变量 w 去掉中间变量剩余的部分称为核心变量.

在这个 ADMM 方法中, 核心变量则是 $v=(y,\lambda)$. y 和 λ 分别是核心变量中的原始部分和对偶部分. 由 (3.6) 生成的序列 $\{v^k=(y^k,\lambda^k)\}$ 满足

$$\|v^{k+1}-v^*\|^2_H\le\|v^k-v^*\|^2_H-\|v^k-v^{k+1}\|^2_H, \tag{3.7}$$

其中

$$v=\begin{pmatrix} y\\ \lambda\end{pmatrix},\qquad H=\begin{pmatrix}\beta B^TB & 0\\ 0 & \frac{1}{\beta}I\end{pmatrix}.$$

不等式 (3.7) 表明核心变量序列 $\{v^k=(y^k,\lambda^k)\}$ 收缩, 它是保证乘子交替方向法收敛的关键不等式. 一个比较简单的证明可以参考我主页上系列讲义 [13] 的第 11 讲. 证明与 §2.2 中证明增广 Lagrange 乘子法一样, 利用变分不等式为工具, 得到与 (2.5), (2.7) 相仿 (但不尽相同) 的式子, 最后得到 (3.7).

ADMM 被认为是求解含两个可分离算子的凸优化问题非常有效的方法 [1], 近年来在图像处理, 稀疏优化中发现大量应用. 算法可以解释为

核心变量的原始部分 $\boldsymbol{y}$ 跟核心变量的对偶部分 $\boldsymbol{\lambda}$ 对弈.

ADMM 收敛的原因是核心变量 $v=(y,\lambda)$ 中的原始部分 y 和对偶部分 λ 是对等的. 我们证明了 ADMM 同时具有遍历意义和非遍历意义下 $O(1/t)$ 的收敛性质 [10, 11].

4　三个可分离算子的凸优化问题

我们继续考虑三个可分离算子的线性约束凸优化问题

$$\min\{\theta_1(x)+\theta_2(y)+\theta_3(z) \mid Ax+By+Cz=b,\, x\in\mathcal{X}, y\in\mathcal{Y}, z\in\mathcal{Z}\}. \tag{4.1}$$

它的 Lagrange 函数是

$$L^3(x,y,z,\lambda)=\theta_1(x)+\theta_2(y)+\theta_3(z)-\lambda^T(Ax+By+Cz-b). \tag{4.2}$$

4.1　Lagrange 函数 (4.2) 鞍点等价的变分不等式

与前面同样的分析告知我们, Lagrange 函数 (4.2) 的鞍点$\big((x^*,y^*,z^*),\lambda^*\big)$ 是变分不等式

$$w^*\in\Omega, \quad \theta(u)-\theta(u^*)+(w-w^*)^TF(w^*)\ge 0, \quad \forall\, w\in\Omega, \tag{4.3a}$$

的解, 其中

$$w=\begin{pmatrix} x\\ y\\ z\\ \lambda\end{pmatrix}, \quad u=\begin{pmatrix} x\\ y\\ z\end{pmatrix}, \quad F(w)=\begin{pmatrix} -A^T\lambda\\ -B^T\lambda\\ -C^T\lambda\\ Ax+B+Cz-b\end{pmatrix}, \tag{4.3b}$$

$$\theta(u)=\theta_1(x)+\theta_2(y)+\theta_3(z), \qquad \Omega=\mathcal{X}\times\mathcal{Y}\times\mathcal{Z}\times\Re^m.$$

凸优化问题 (4.1) 的的增广 Lagrange 函数由它的 Lagrange 函数 $L^3(x,y,z,\lambda)$ (4.2) 和等式线性约束的二次函数 $\frac{\beta}{2}\|Ax+By+Cz-b\|^2$ 组成, 即

$$\mathcal{L}^3_\beta(x,y,z,\lambda)=\theta_1(x)+\theta_2(y)+\theta_3(z)-\lambda^T(Ax+By+Cz-b)+\frac{\beta}{2}\|Ax+By+Cz-b\|^2. \tag{4.4}$$

如果我们不将 y 和 z 分离, 而是把 (y,z) 捆在一起, 采用 ADMM (3.6) 去求解, 迭代公式就是

$$\left\{\begin{array}{rcl} x^{k+1} &=& \arg\min\big\{\mathcal{L}^3_\beta(x,y^k,z^k,\lambda^k)\;\big|\; x\in\mathcal{X}\big\},\\ (y^{k+1},z^{k+1}) &=& \arg\min\big\{\mathcal{L}^3_\beta(x^{k+1},y,z,\lambda^k)\;\big|\; y\in\mathcal{Y}, z\in\mathcal{Z}\big\},\\ \lambda^{k+1} &=& \lambda^k-\beta\big(Ax^{k+1}+By^{k+1}+Cz^{k+1}-b\big). \end{array}\right. \tag{4.5}$$

这样方法是收敛的. **缺点** 是原问题本来具备的 y 和 z 的可分离性质没有得到充分利用！

4.2　直接推广的 ADMM 对三个可分离算子的问题不能保证收敛

人们自然想到直接推广 ADMM (**Direct Extension of ADMM**) 来求解三个可分离算子的问题. 如果我们用直接推广的乘子交替方向法求解问题 (4.1), k-次迭代从一个给定的 (y^k,z^k,λ^k) 开始, 通过

$$\left\{\begin{array}{l} x^{k+1}=\arg\min\big\{\mathcal{L}^3_\beta(x,y^k,z^k,\lambda^k)\;\big|\; x\in\mathcal{X}\big\},\\ y^{k+1}=\arg\min\big\{\mathcal{L}^3_\beta(x^{k+1},y,z^k,\lambda^k)\;\big|\; y\in\mathcal{Y}\big\},\\ z^{k+1}=\arg\min\big\{\mathcal{L}^3_\beta(x^{k+1},y^{k+1},z,\lambda^k)\;\big|\; z\in\mathcal{Z}\big\},\\ \lambda^{k+1}=\lambda^k-\beta\big(Ax^{k+1}+By^{k+1}+Cz^{k+1}-b\big), \end{array}\right. \tag{4.6}$$

给出新的 $(y^{k+1},z^{k+1},\lambda^{k+1})$. **就像机械图纸, 为有讨论的共同语言, 我们建议只把** (4.6) **称做 ADMM 的直接推广.** 在任何一点与 (4.6) 不同的, 都是某种修正的格式.

相当长的一段时间内, 人们不知道 (4.6) 是否收敛. 直觉让我们认为不能保证收敛. 原因是, 用 (4.6) 处理三个算子的问题, 采用相应的术语, 核心变量 $v=(y,z,\lambda)$, 其中

$\boldsymbol{(y,z)}$ **是核心变量的原始部分 (primal part),** $\boldsymbol{\lambda}$ **是核心变量的对偶部分 (dual part)**.

问题是对核心变量的原始部分中的 y 和 z 的两部分, 算法 (4.6) 是不公平的**(unfair)**.

- 在求解 y-子问题时, 只能用 $(x^{k+1}, z^k, \lambda^k)$ 的信息, 因为 z^k 尚未更新;
- 在求解 z-子问题时, 因为新的 y^{k+1} 已经有了, 就用 $(x^{k+1}, y^{k+1}, \lambda^k)$ 的信息.

用直接推广的 ADMM 来求解三个可分离算子的问题 (4.1), 很多实际计算还是成功的. 但是, **犹如一个家长带了这两个孩子 (核心变量中的原始部分 y 和 z) 去跟人家(核心变量中的对偶部分 λ) 对弈, 家长对自己的两个孩子(y 和z) 就不公平, 难免那个意识到不公的孩子消极怠工, 影响终极目标的实现.**

在一时也没有举出不收敛的例子的那段时间, 我们针对算法 (4.6) 内在的不公平性, 先做了一些修正的 ADMM [7, 8], 用来处理多个算子的问题. 论文 [7] 2012 年上半年就在 SIAM J. Optim. 发表, 同期撰写的 [8], 2015 年初才在 IMA Numer. Analysis 发表, 都受到了一定的关注. 这些方法我们将在下一节中具体介绍.

我们当然希望发表在论文 [7, 8] 中的修正方法是必要的, 所以一直想从最简单的线性方程组着手, 证明方法 (4.6) 不能保证收敛. 直到 2013 年下半年, 经过多人合作努力, 借助计算机, 才对三个或者三个以上算子的问题找到了不收敛的例子 [2].

将求解一个线性方程组作为问题 (4.1) **的特例**. 问题是:

$$\begin{array}{rl} \min & 0\cdot x + 0\cdot y + 0\cdot z \\ \text{s.t} & Ax + By + Cz = 0. \end{array} \qquad \text{其中} \qquad (A,B,C) = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 2 \\ 1 & 2 & 2 \end{pmatrix}. \tag{4.7}$$

由于矩阵 (A,B,C) 非奇异, 问题 (4.7) 的惟一解是

$$\begin{pmatrix} x^* \\ y^* \\ z^* \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad \text{相应的最优乘子} \quad \lambda^* = \begin{pmatrix} \lambda_1^* \\ \lambda_2^* \\ \lambda_3^* \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

用直接推广的ADMM (4.6) 求解这个问题, 核心变量是 v, $v^T = (y, z, \lambda_1, \lambda_2, \lambda_3)$. 迭代公式可以化为

$$v^{k+1} = Mv^k, \qquad \text{其中} \qquad M = \frac{1}{162}\begin{pmatrix} 144 & -9 & -9 & -9 & 18 \\ 8 & 157 & -5 & 13 & -8 \\ 64 & 122 & 122 & -58 & -64 \\ 56 & -35 & -35 & 91 & -56 \\ -88 & -26 & -26 & -62 & 88 \end{pmatrix}. \tag{4.8}$$

因此, 我们就有

$$v^{k+1} = Mv^k = \cdots = M^{k+1}v^0.$$

对 (4.8) 中的矩阵 M, 恰有 $\rho(M) > 1$. 可以选一个非零的 v^0 开始, 采用上述迭代, 就有 $\lim_{k\to\infty} \|v^k\| = +\infty$. 说明算法 (4.6) 对一般的三个算子的问题 (4.1) 不能保证收敛. 实际计算中, 我们从任何一个随机的 $v^0 \neq 0 \in \Re^5$ 出发, 用 (4.8) 迭代, 都有 $\lim_{k\to\infty} \|v^k\| = +\infty$.

5 改造 ADMM 成求解三个算子问题的收敛方法

既然算法 (4.6) 对一般的三个算子的问题 (4.1) 不能保证收敛的原因是它在求解子问题时对原始核心变量中的 y 和 z 的两部分不公平, 因此, 我们考虑改造 ADMM 成求解三个算子问题的收敛方法的路, 就自然成了以下两条:

1. 将 (4.6) 提供的 y^{k+1} 和 z^{k+1} 改造, 对 (4.6) 造成的不公平进行适当找补;
2. 对算法 (4.6) 本身进行改造, 使得它处理原始核心变量中 y 和 z 两部分时公平.

5.1 找补校正的方法

第一条路子, 是对 (4.6) 生成的 y^{k+1} 和 z^{k+1} 进行找补(校正). 二次分配中能让不公平变公平的方法很多, 这里的校正通过

$$\begin{pmatrix} y^{k+1} \\ z^{k+1} \end{pmatrix} := \begin{pmatrix} y^{k} \\ z^{k} \end{pmatrix} - \nu \begin{pmatrix} I & -(B^TB)^{-1}B^TC \\ 0 & I \end{pmatrix} \begin{pmatrix} y^{k} - y^{k+1} \\ z^{k} - z^{k+1} \end{pmatrix}, \quad \nu \in (0,1] \tag{5.1}$$

实现. 这里用赋值号 (:=), 表示 (5.1) 右端的 (y^{k+1}, z^{k+1}) 是直接推广的乘子交替方向法 (4.6) 提供的.

注意到, 略去子问题中的常数项, 直接推广的交替方向法 (4.6) 可以写成它的等价形式

$$\begin{cases} x^{k+1} = \arg\min\{\theta_1(x) - (\lambda^k)^T(Ax) + \frac{\beta}{2}\|Ax + By^k + Cz^k - b\|^2 \mid x \in \mathcal{X}\}, \\ y^{k+1} = \arg\min\{\theta_2(y) - (\lambda^k)^T(By) + \frac{\beta}{2}\|Ax^{k+1} + By + Cz^k - b\|^2 \mid y \in \mathcal{Y}\}, \\ z^{k+1} = \arg\min\{\theta_3(z) - (\lambda^k)^T(Cz) + \frac{\beta}{2}\|Ax^{k+1} + By^{k+1} + Cz - b\|^2 \mid z \in \mathcal{Z}\}, \\ \lambda^{k+1} = \lambda^k - \beta(Ax^{k+1} + By^{k+1} + Cz^{k+1} - b). \end{cases} \tag{5.2}$$

从 (5.2) 可以看出, 开始执行 k-次迭代需要的是 (By^k, Cz^k, λ^k). 因此, 为下一次迭代我们也只要提供 $(By^{k+1}, Cz^{k+1}, \lambda^{k+1})$. 由 (5.1), 这组 (By^{k+1}, Cz^{k+1}) 可以由

$$\begin{pmatrix} By^{k+1} \\ Cz^{k+1} \end{pmatrix} := \begin{pmatrix} By^{k} \\ Cz^{k} \end{pmatrix} - \nu \begin{pmatrix} I & -I \\ 0 & I \end{pmatrix} \begin{pmatrix} B(y^{k} - y^{k+1}) \\ C(z^{k} - z^{k+1}) \end{pmatrix}, \quad \nu \in (0,1] \tag{5.3}$$

提供, 右端的 (y^{k+1}, z^{k+1}) 是由 (4.6) 提供的. 这样, 我们就无需计算 $(B^TB)^{-1}$. 这是变动最小、最简单的一种校正. 特别地, 当 $\nu = 1$, 就只要对 By 进行校正, 公式就简化成

$$By^{k+1} := By^{k+1} + C(z^k - z^{k+1}). \tag{5.4}$$

采用 $\nu = 1$, 不能保证每步都(向解集靠拢)收缩, 只能保证在遍历意义下(所有迭代点的算术平均向解集)收敛. 在实际计算中[7], 我们取 $\nu = [0.9, 0.95]$, 称这个方法为**《带高斯回代的交替方向法》(Alternating direction method with Gaussian back substitution)**.

为什么要在 (5.1) 的右端取个小于 1 的常数 ν 呢? 注意到 (4.5)是能保证收敛的, 其中

$$\text{核心变量的原始部分} \quad (y, z) \quad \text{跟核心变量的对偶部分} \quad \lambda \quad \text{对弈}.$$

从 (4.5) 到 (4.6), 松弛了的是核心变量中原始部分 (y 和 z) 的子问题求解. 所以, 在总体找补的时候, 加个适当的缩小量 $\nu \in [0.9, 0.95]$ 是合理的.

5.2 强制平等的方法

按照第二条路子, 对算法 (4.6) 进行改造, 使得它公平处理原始核心变量中 y 和 z 两部分. 一个简单的想法就是在求解 z-子问题时, 即使 y^{k+1} 已经有了, 我们还是只用 y^k 的信息. 这样就导致了下面的算法:

$$\begin{cases} x^{k+1} = \arg\min\big\{\mathcal{L}^3_\beta(x, y^k, z^k, \lambda^k) \;\big|\; x \in \mathcal{X}\big\}, \\ y^{k+1} = \arg\min\big\{\mathcal{L}^3_\beta(x^{k+1}, y, z^k, \lambda^k) \;\big|\; y \in \mathcal{Y}\big\}, \\ z^{k+1} = \arg\min\big\{\mathcal{L}^3_\beta(x^{k+1}, y^k, z, \lambda^k) \;\big|\; z \in \mathcal{Z}\big\}, \\ \lambda^{k+1} = \lambda^k - \beta\big(Ax^{k+1} + By^{k+1} + Cz^{k+1} - b\big). \end{cases}$$

与收敛的方法 (4.5) 比较, 相当于将 (4.5) 中的

$$(y^{k+1}, z^{k+1}) = \arg\min\big\{\mathcal{L}^3_\beta(x^{k+1}, y, z, \lambda^k) \;\big|\; y \in \mathcal{Y}, z \in \mathcal{Z}\big\},$$

改成了

$$\begin{cases} y^{k+1} = \arg\min\big\{\mathcal{L}_\beta^3(x^{k+1}, y, z^k, \lambda^k) \;\big|\; y \in \mathcal{Y}\big\}, \\ z^{k+1} = \arg\min\big\{\mathcal{L}_\beta^3(x^{k+1}, y^k, z, \lambda^k) \;\big|\; z \in \mathcal{Z}\big\}. \end{cases}$$

这样做, 原问题的可分离的性质利用了, 对 y 和 z 也公平了, 但是也有点太自由化了. 想放松就放松, 显然是不行的, 也可以举出这样做不收敛的例子.

怎么办? 我们在平等平行处理 y 和 z 子问题的时候, 目标函数后面都额外再加一个正则项, 迭代法变成

$$\begin{cases} x^{k+1} = \arg\min\{\mathcal{L}_\beta^3(x, y^k, z^k, \lambda^k) \,|\, x \in \mathcal{X}\}, \\ y^{k+1} = \arg\min\{\mathcal{L}_\beta^3(x^{k+1}, y, z^k, \lambda^k) + \frac{\tau\beta}{2}\|B(y-y^k)\|^2 \,|\, y \in \mathcal{Y}\}, \\ z^{k+1} = \arg\min\{\mathcal{L}_\beta^3(x^{k+1}, y^k, z, \lambda^k) + \frac{\tau\beta}{2}\|C(z-z^k)\|^2 \,|\, z \in \mathcal{Z}\}, \\ \lambda^{k+1} = \lambda^k - \beta(Ax^{k+1} + By^{k+1} + Cz^{k+1} - b), \end{cases} \tag{5.5}$$

其中 $\tau > 1$. 加正则项的意义就是在**自由化**的情况下再加上个**自我约束机制**.

略去算法 (5.5) 中 y 和 z 子问题的常数项, 方法也可以改写成等价的

$$\begin{cases} x^{k+1} = \arg\min\{\mathcal{L}_\beta^3(x, y^k, z^k, \lambda^k) \,|\, x \in \mathcal{X}\}, \\ \lambda^{k+\frac{1}{2}} = \lambda^k - \beta(Ax^{k+1} + By^k + Cz^k - b), \\ y^{k+1} = \arg\min\{\theta_2(y) - (\lambda^{k+\frac{1}{2}})^T By + \frac{\mu\beta}{2}\|B(y-y^k)\|^2 \,|\, y \in \mathcal{Y}\}, \\ z^{k+1} = \arg\min\{\theta_3(z) - (\lambda^{k+\frac{1}{2}})^T Cz + \frac{\mu\beta}{2}\|C(z-z^k)\|^2 \,|\, z \in \mathcal{Z}\}, \\ \lambda^{k+1} = \lambda^k - \beta(Ax^{k+1} + By^{k+1} + Cz^{k+1} - b), \end{cases} \tag{5.6}$$

其中 $\mu = \tau + 1 > 2$. 对这两个算法等价性有兴趣的读者, 只要验证它们 y 和 z-子问题的最优性条件相同就是了. 我们在 2010 年就提出了算法 (5.6), 随即被 UCLA 教授 Standy Osher 的课题组当年就在论文 [3] 中有超过半页的实质引用. 论文处理的非负矩阵分解和降维问题, 其数学模型就是 (4.1). 文中称我们的方法为 ADMM-Like Method.

这一节提到的对三个算子问题改造过的 ADMM 类算法, 是 [7, 8] 中处理多个算子方法的特例. 详细的收敛性证明可以查阅我主页上的报告 [12].

6 统一框架下的算法描述和收敛条件

我们将要求解的问题都看成形如 (4.3) 的变分不等式, 也就是

$$w^* \in \Omega, \quad \theta(u) - \theta(u^*) + (w - w^*)^T F(w^*) \ge 0, \quad \forall\, w \in \Omega.$$

核心变量是 v, 是在 w 中去掉中间变量 x 以后的部分分量. 把算法都归结为如下的预测-校正方法的统一框架:

[预测.] 对给定的 v^k，求得一个 $\tilde{w}^k$, 使得

$$\tilde{w}^k \in \Omega, \;\; \theta(u) - \theta(\tilde{u}^k) + (w - \tilde{w}^k)^T F(\tilde{w}^k) \ge (v - \tilde{v}^k)^T Q(v^k - \tilde{v}^k), \; \forall w \in \Omega, \tag{6.1a}$$

其中 Q 不一定对称, 但是要求 $Q^T + Q$ 正定.

[校正.] 给出新迭代点 v^{k+1} 的校正公式为

$$v^{k+1} = v^k - M(v^k - \tilde{v}^k). \tag{6.1b}$$

将收敛性条件也归纳成下面两条

收敛性条件
对算法 (6.1) 中的矩阵 Q 和 M, 有正定矩阵 H, 使得

$$HM = Q. \tag{6.2a}$$

并且

$$G = Q^T + Q - M^T HM \succ 0, \quad (\text{至少半正定}). \tag{6.2b}$$

在这个统一框架和相应的收敛性条件下, 能够帮助我们构造新的收敛算法 [6], 证明收敛性就变得特别容易, 收敛速率的分析也迎刃而解. 这一节只验证 §5 中修正的方法, 可以纳入算法框架 (6.1), 并且满足收敛性条件 (6.2). 证明放到 §7 中去陈述.

6.1 找补校正的方法

我们用 (5.2) 的输出$x^{k+1}, y^{k+1}, z^{k+1}$ 定义辅助的预测变量 $\tilde{w}^k = (\tilde{x}^k, \tilde{y}^k, \tilde{z}^k, \tilde{\lambda}^k)$, 其中

$$(\tilde{x}^k, \tilde{y}^k, \tilde{z}^k) = (x^{k+1}, y^{k+1}, z^{k+1}), \tag{6.3a}$$

$$\tilde{\lambda}^k = \lambda^k - \beta(A\tilde{x}^k + By^k + Cz^k - b). \tag{6.3b}$$

根据优化问题的最优性条件 (1.1), 用 $\tilde{x}^k = x^{k+1}$ 和 (6.3b), 可以将 (5.2) 中 x-子问题的最优性条件写成

$$\tilde{x}^k \in \mathcal{X}, \;\; \theta_1(x) - \theta_1(\tilde{x}^k) + (x - \tilde{x}^k)^T(-A^T\tilde{\lambda}^k) \ge 0, \;\; \forall x \in \mathcal{X}. \tag{6.4a}$$

根据同样的道理, 对 y 和 z-子问题的最优性条件, 分别有

$$\tilde{y}^k \in \mathcal{Y}, \;\; \theta_2(y) - \theta_2(\tilde{y}^k) + (y - \tilde{y}^k)^T(-B^T\tilde{\lambda}^k) \ge (y - \tilde{y}^k)^T\beta B^T B(y^k - \tilde{y}^k), \;\; \forall y \in \mathcal{Y}. \tag{6.4b}$$

$$\begin{aligned} \tilde{z}^k \in \mathcal{Z}, \qquad & \theta_3(z) - \theta_3(\tilde{z}^k) + (z - \tilde{z}^k)^T\big(-C^T\tilde{\lambda}^k\big) \\ & \ge (z - \tilde{z}^k)^T\big(\beta C^T B(y^k - \tilde{y}^k) + \beta C^T C(z^k - \tilde{z}^k)\big), \quad \forall z \in \mathcal{Z}. \end{aligned} \tag{6.4c}$$

根据关系式 (6.3b), 等式

$$(A\tilde{x}^k + B\tilde{y}^k + C\tilde{z}^k - b) - B(\tilde{y}^k - y^k) - C(\tilde{z}^k - z^k) + \tfrac{1}{\beta}(\tilde{\lambda}^k - \lambda^k) = 0$$

可以写成变分不等式的形式

$$\begin{aligned} & \tilde{\lambda}^k \in \Re^m, \;\; (\lambda - \tilde{\lambda}^k)^T(A\tilde{x}^k + B\tilde{y}^k + C\tilde{z}^k - b) \\ & \quad \ge \;\; (\lambda - \tilde{\lambda}^k)^T\big\{-B(y^k - \tilde{y}^k) - C(z^k - \tilde{z}^k) + \tfrac{1}{\beta}(\lambda^k - \tilde{\lambda}^k)\big\}, \; \forall \lambda \in \Re^m. \end{aligned} \tag{6.4d}$$

利用变分不等式 (4.3), 我们可以将 (6.4) 中的结果表述成下面的引理.

Lemma 6.1. 设 $(x^{k+1}, y^{k+1}, z^{k+1})$ 根据给定的 v^k 由 (5.2) 生成. 那么, 由 (6.3) 定义的预测点 $\tilde{w}^k$ 满足

$$\tilde{w}^k \in \Omega, \;\; \theta(u) - \theta(\tilde{u}^k) + (w - \tilde{w}^k)^T F(\tilde{w}^k) \ge (v - \tilde{v}^k)^T Q(v^k - \tilde{v}^k), \; \forall\, w \in \Omega, \tag{6.5}$$

其中

$$Q = \begin{pmatrix} \beta B^T B & 0 & 0 \\ \beta C^T B & \beta C^T C & 0 \\ -B & -C & \frac{1}{\beta} I_m \end{pmatrix}. \tag{6.6}$$

我们接着来给出校正矩阵. 首先, 由定义 (6.3a), 对 y 和 z 进行校正的公式 (5.1) 则是

$$\begin{pmatrix} y^{k+1} \\ z^{k+1} \end{pmatrix} = \begin{pmatrix} y^k \\ z^k \end{pmatrix} - \begin{pmatrix} \nu I & -\nu(B^TB)^{-1}B^TC \\ 0 & \nu I \end{pmatrix} \begin{pmatrix} y^k - \tilde{y}^k \\ z^k - \tilde{z}^k \end{pmatrix}. \tag{6.7a}$$

另外, 用 $(\tilde{y}^k, \tilde{z}^k, \tilde{\lambda}^k)$, 那个由 (5.2) 生成, 无需校正的 λ^{k+1} 可以写成

$$\begin{aligned} \lambda^{k+1} &= \lambda^k - \beta\big(Ax^{k+1} + By^{k+1} + Cz^{k+1} - b\big) \\ &= \lambda^k - \big[-\beta B(y^k - \tilde{y}^k) - \beta C(z^k - \tilde{z}^k) + (\lambda^k - \tilde{\lambda}^k)\big] \end{aligned} \tag{6.7b}$$

因此, 将 (6.7) 写在一起就是

$$v^{k+1} = v^k - M(v^k - \tilde{v}^k),$$

其中

$$M = \begin{pmatrix} \nu I & -\nu(B^TB)^{-1}B^TC & 0 \\ 0 & \nu I & 0 \\ -\beta B & -\beta C & I_m \end{pmatrix}. \tag{6.8}$$

这样, 我们对 §5.1 中的找补校正方法, 纳入了预测-校正方法框架 (6.1).

最后我们来验证收敛性条件. 首先, 矩阵 Q 可以写成分块的形式

$$Q = \begin{pmatrix} \beta Q_0 & 0 \\ -\mathcal{A} & \frac{1}{\beta} I_m \end{pmatrix}, \quad 其中 \quad Q_0 = \begin{pmatrix} B^TB & 0 \\ C^TB & \beta C^TC \end{pmatrix}, \quad \mathcal{A} = (B, C).$$

我们记

$$D_0 = \begin{pmatrix} B^TB & 0 \\ 0 & C^TC \end{pmatrix} \qquad 则有 \qquad Q_0^T + Q_0 = D_0 + \mathcal{A}^T\mathcal{A}.$$

利用这些符号, 对 (6.8) 中的矩阵 M, 就可以写成

$$M = \begin{pmatrix} \nu Q_0^{-T} D_0 & 0 \\ -\beta\mathcal{A} & I \end{pmatrix} \qquad 它的逆矩阵是 \qquad M^{-1} = \begin{pmatrix} \frac{1}{\nu} D_0^{-1} Q_0^T & 0 \\ \frac{1}{\nu}\beta\mathcal{A} D_0^{-1} Q_0^T & I \end{pmatrix}. \tag{6.9}$$

要问是否有正定矩阵 H 使得 $HM = Q$, 只要验证矩阵

$$H = QM^{-1} = \begin{pmatrix} \beta Q_0 & 0 \\ -\mathcal{A} & \frac{1}{\beta} I_m \end{pmatrix} \begin{pmatrix} \frac{1}{\nu} D_0^{-1} Q_0^T & 0 \\ \frac{1}{\nu}\beta\mathcal{A} D_0^{-1} Q_0^T & I \end{pmatrix} = \begin{pmatrix} \frac{1}{\nu}\beta Q_0 D_0^{-1} Q_0^T & 0 \\ 0 & \frac{1}{\beta} I \end{pmatrix}$$

是否正定, 确实, 矩阵 H 是正定的. 进而我们验证矩阵 G 的正定性.

根据 (6.2b) 对矩阵 G 的定义, 利用 $HM = Q$ 和 $Q_0^T + Q_0 = D_0 + \mathcal{A}^T\mathcal{A}$, 就会得到

$$\begin{aligned} G &= Q^T + Q - M^THM = Q^T + Q - Q^TM \\ &= (Q^T + Q) - \begin{pmatrix} \beta Q_0^T & -\mathcal{A}^T \\ 0 & \frac{1}{\beta} I_m \end{pmatrix} \begin{pmatrix} \nu Q_0^{-T} D_0 & 0 \\ -\beta\mathcal{A} & I \end{pmatrix} \\ &= \begin{pmatrix} \beta(Q_0^T + Q_0) & -\mathcal{A}^T \\ -\mathcal{A} & \frac{2}{\beta} I_m \end{pmatrix} - \begin{pmatrix} \beta(\nu D_0 + \mathcal{A}^T\mathcal{A}) & -\mathcal{A}^T \\ -\mathcal{A} & \frac{2}{\beta} I_m \end{pmatrix} \\ &= \begin{pmatrix} (1-\nu)\beta D_0 & 0 \\ 0 & \frac{1}{\beta} I_m \end{pmatrix} \succ 0, \qquad (因为\ \nu \in (0,1)). \end{aligned}$$

至此, 证明了找补校正的方法纳入框架 (6.1) 后, 相应的收敛性条件 (6.2) 是也得到满足的.

最后我们想说一下, 为什么这个方法叫做 ADMM with Gaussian back Substitution 呢? 利用 (6.9) 中矩阵 M 的表达式, 校正 (6.7a) 可以写成

$$\begin{pmatrix} y^{k+1} \\ z^{k+1} \end{pmatrix} = \begin{pmatrix} y^k \\ z^k \end{pmatrix} - \nu Q_0^{-T} D_0 \begin{pmatrix} y^k - \tilde{y}^k \\ z^k - \tilde{z}^k \end{pmatrix}.$$

实际计算中, 它是通过

$$Q_0^T \begin{pmatrix} y^{k+1} - y^k \\ z^{k+1} - z^k \end{pmatrix} = \nu D_0 \begin{pmatrix} \tilde{y}^k - y^k \\ \tilde{z}^k - z^k \end{pmatrix} \tag{6.10}$$

来实现, 注意到 (6.10) 中的矩阵 Q_0^T 是一个上三角矩阵, 这样先后求得 z^{k+1}, y^{k+1} 的过程我们把它说成是高斯回代.

6.2 强制平等的方法

为了利用统一框架 (6.1) 及其相应的收敛性条件 (6.2), 我们以 (5.6) 的输出 $x^{k+1}, y^{k+1}, z^{k+1}$ 定义辅助的预测变量 $\tilde{w}^k = (\tilde{x}^k, \tilde{y}^k, \tilde{z}^k, \tilde{\lambda}^k)$, 其中

$$(\tilde{x}^k, \tilde{y}^k, \tilde{z}^k) = (x^{k+1}, y^{k+1}, z^{k+1}), \tag{6.11a}$$

$$\tilde{\lambda}^k = \lambda^k - \beta(A\tilde{x}^k + By^k + Cz^k - b). \tag{6.11b}$$

根据优化问题的最优性条件 (1.1), 用 $\tilde{x}^k = x^{k+1}$ 和 (6.11b), 可以将 (5.6) 中 x-子问题的最优性条件写成

$$\tilde{x}^k \in \mathcal{X}, \;\; \theta_1(x) - \theta_1(\tilde{x}^k) + (x - \tilde{x}^k)^T(-A^T\tilde{\lambda}^k) \ge 0, \;\; \forall x \in \mathcal{X}. \tag{6.12a}$$

由于 $\tilde{\lambda}^k = \lambda^{k+\frac{1}{2}}$, (5.6) 中 y 和 z 子问题是平行求解的, 它们的最优性条件分别有

$$\tilde{y}^k \in \mathcal{Y}, \;\; \theta_2(y) - \theta_2(\tilde{y}^k) + (y - \tilde{y}^k)^T(-B^T\tilde{\lambda}^k) \ge (y - \tilde{y}^k)^T \mu\beta B^T B(y^k - \tilde{y}^k), \; \forall y \in \mathcal{Y}. \tag{6.12b}$$

和

$$\tilde{z}^k \in \mathcal{Z}, \;\; \theta_3(z) - \theta_3(\tilde{z}^k) + (z - \tilde{z}^k)^T(-C^T\tilde{\lambda}^k) \ge (z - \tilde{z}^k)^T \mu\beta C^T C(\tilde{z}^k - z^k), \; \forall z \in \mathcal{Z}. \tag{6.12c}$$

同样, 由等式 (6.11b) 得到

$$\begin{aligned} &\tilde{\lambda}^k \in \Re^m, \;\; (\lambda^k - \tilde{\lambda}^k)^T(A\tilde{x}^k + B\tilde{y}^k + C\tilde{z}^k - b) \\ &\quad \ge \;\; (\lambda^k - \tilde{\lambda}^k)^T\big\{-B(y^k - \tilde{y}^k) - C(z^k - \tilde{z}^k) + \tfrac{1}{\beta}(\lambda^k - \tilde{\lambda}^k)\big\}, \; \forall \lambda \in \Re^m. \end{aligned} \tag{6.12d}$$

利用变分不等式 (4.3), 我们可以将 (6.12) 中的结果表述成下面的引理.

Lemma 6.2. 设 $(x^{k+1}, y^{k+1}, z^{k+1})$ 根据给定的 v^k 由 (5.6) 生成. 那么, 由 (6.11) 定义的预测点 $\tilde{w}^k$ 满足

$$\theta(u) - \theta(\tilde{u}^k) + (w - \tilde{w}^k)^T F(\tilde{w}^k) \ge (v - \tilde{v}^k)^T Q(v^k - \tilde{v}^k), \; \forall\, w \in \Omega, \tag{6.13}$$

其中

$$Q = \begin{pmatrix} \mu\beta B^T B & 0 & 0 \\ 0 & \mu\beta C^T C & 0 \\ -B & -C & \frac{1}{\beta} I_m \end{pmatrix}. \tag{6.14}$$

我们接着来找出校正矩阵. 在迭代法 (5.6) 中不含校正, 为了用统一框架证明算法的收敛性, 我们根据 (6.11) 定义了$\tilde{w}^k = (\tilde{x}^k, \tilde{y}^k, \tilde{z}^k, \tilde{\lambda}^k)$. 将这个 $\tilde{w}^k$ 看做预测点, 再找出用 $v^k = (y^k, z^k, \lambda^k)$ 和 $\tilde{v}^k = (\tilde{y}^k, \tilde{z}^k, \tilde{\lambda}^k)$ 表示 $v^{k+1} = (y^{k+1}, z^{k+1}, \lambda^{k+1})$ 的关系式. 由于

$$y^{k+1} = \tilde{y}^k, \qquad z^{k+1} = \tilde{z}^k, \tag{6.15a}$$

和

$$\begin{aligned}\lambda^{k+1} &= \lambda^k - \beta\big(Ax^{k+1} + By^{k+1} + Cz^{k+1} - b\big) \\ &= \lambda^k - \big[-\beta B(y^k - \tilde{y}^k) - \beta C(z^k - \tilde{z}^k) + (\lambda^k - \tilde{\lambda}^k)\big]\end{aligned} \tag{6.15b}$$

这个校正公式可以写成

$$\begin{pmatrix} y^{k+1} \\ z^{k+1} \\ \lambda^{k+1} \end{pmatrix} = \begin{pmatrix} y^k \\ z^k \\ \lambda^k \end{pmatrix} - \begin{pmatrix} I & 0 & 0 \\ 0 & I & 0 \\ -\beta B & -\beta C & I_m \end{pmatrix} \begin{pmatrix} y^k - \tilde{y}^k \\ z^k - \tilde{z}^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix}.$$

换句话说, 我们有

$$v^{k+1} = v^k - M(v^k - \tilde{v}^k), \qquad \text{其中} \qquad M = \begin{pmatrix} I & 0 & 0 \\ 0 & I & 0 \\ -\beta B & -\beta C & I_m \end{pmatrix}. \tag{6.16}$$

这样, 我们对 §5.2 中强制平等的方法, 纳入了预测-校正方法框架 (6.1).

最后我们来验证收敛性条件. 要问是否有正定矩阵 H 使得 $HM = Q$, 只要验证矩阵

$$\begin{aligned} H = QM^{-1} &= \begin{pmatrix} \mu\beta B^TB & 0 & 0 \\ 0 & \mu\beta C^TC & 0 \\ -B & -C & \frac{1}{\beta}I_m \end{pmatrix} \begin{pmatrix} I & 0 & 0 \\ 0 & I & 0 \\ \beta B & \beta C & I_m \end{pmatrix} \\ &= \begin{pmatrix} \mu\beta B^TB & 0 & 0 \\ 0 & \mu\beta C^TC & 0 \\ 0 & 0 & \frac{1}{\beta}I \end{pmatrix}, \end{aligned} \tag{6.17}$$

上式表示矩阵 H 是正定的. 进而我们验证矩阵 G 的正定性. 根据定义和 $Q = HM$, 有

$$\begin{aligned} G &= Q^T + Q - M^THM = Q^T + Q - Q^TM \\ &= \begin{pmatrix} 2\mu\beta B^TB & 0 & -B^T \\ 0 & 2\mu\beta C^TC & -C^T \\ -B & -C & \frac{2}{\beta}I_m \end{pmatrix} - \begin{pmatrix} \mu\beta B^TB & 0 & -B^T \\ 0 & \mu\beta C^TC & -C^T \\ 0 & 0 & \frac{1}{\beta}I_m \end{pmatrix} \begin{pmatrix} I & 0 & 0 \\ 0 & I & 0 \\ -\beta B & -\beta C & I_m \end{pmatrix}. \end{aligned}$$

继续进行一些基本的矩阵运算就能得到

$$G = \begin{pmatrix} (\mu-2)\beta B^TB & 0 & 0 \\ 0 & (\mu-2)\beta C^TC & 0 \\ 0 & 0 & \frac{1}{\beta}I \end{pmatrix} + \beta \begin{pmatrix} B^TB & -B^TC & 0 \\ -C^TB & C^TC & 0 \\ 0 & 0 & 0 \end{pmatrix}. \tag{6.18}$$

上式右端的最后一个矩阵半正定. 所以, 由于 $\mu > 2$, 矩阵 G 是正定的. 我们把 §5.2 中的强制平等的方法, 纳入了预测-校正方法框架 (6.1) 后, 证明了它满足收敛性条件 (6.2).

7 收缩性质和收敛速率的证明

对形如 (4.3) 的变分不等式, 这一节利用统一框架 (6.1) 和条件 (6.2) 证明收敛性.

Lemma 7.1. 求解变分不等式 (4.3), 如果条件 (6.2) 满足, 由算法 (6.1) 生成的点满足

$$\begin{aligned}\tilde{w}^k \in \Omega,\ \ \theta(u) - \theta(\tilde{u}^k) + (w - \tilde{w}^k)^T F(\tilde{w}^k) \\ \ge\ \frac{1}{2}\big(\|v - v^{k+1}\|_H^2 - \|v - v^k\|_H^2\big) + \frac{1}{2}\|v^k - \tilde{v}^k\|_G^2,\ \ \forall w \in \Omega. \qquad (7.1)\end{aligned}$$

证明. 由 (6.1b) 生成新的迭代点. 由于 $Q = HM$, 这时 (6.1a) 就可以改写成

$$\tilde{w}^k \in \Omega,\ \ \theta(u) - \theta(\tilde{u}^k) + (w - \tilde{w}^k)^T F(\tilde{w}^k) \ge (v - \tilde{v}^k)^T H(v^k - v^{k+1}),\ \forall w \in \Omega, \qquad (7.2)$$

对 (7.2) 式右端的 $(v - \tilde{v}^k)^T H(v^k - v^{k+1})$, 用恒等式

$$(a - b)^T H(c - d) = \frac{1}{2}\big(\|a - d\|_H^2 - \|a - c\|_H^2\big) + \frac{1}{2}\big(\|c - b\|_H^2 - \|d - b\|_H^2\big), \qquad (7.3)$$

进行处理, 就有

$$\begin{aligned}&\theta(u) - \theta(\tilde{u}^k) + (w - \tilde{w}^k)^T F(\tilde{w}^k) \\ &\ge\ \frac{1}{2}\big(\|v - v^{k+1}\|_H^2 - \|v - v^k\|_H^2\big) + \frac{1}{2}\big(\|v^k - \tilde{v}^k\|_H^2 - \|v^{k+1} - \tilde{v}^k\|_H^2\big),\ \forall w \in \Omega. \quad (7.4)\end{aligned}$$

对 (7.4) 右端的最后一部分, 利用 (6.1b), $HM = Q$ 和(6.2b) 中对矩阵 G 的定义, 我们有

$$\begin{aligned}&\frac{1}{2}\big(\|v^k - \tilde{v}^k\|_H^2 - \|v^{k+1} - \tilde{v}^k\|_H^2\big) = \frac{1}{2}\big(\|v^k - \tilde{v}^k\|_H^2 - \|(v^k - \tilde{v}^k) - M(v^k - \tilde{v}^k)\|_H^2\big) \\ &=\ \frac{1}{2}\big((v^k - \tilde{v}^k)^T(HM + M^T H - M^T HM)(v^k - \tilde{v}^k)\big) = \frac{1}{2}\|v^k - \tilde{v}^k\|_G^2.\end{aligned}$$

将上述结果代入 (7.4) 右端, 就得到 (7.1), 引理 7.1 得证. □

Theorem 7.1. 对变分不等式问题 (4.3), 由算法框架 (6.1) 生成的核心变量序列 $\{v^k\}$, 在条件 (6.2) 满足的时候, 有收缩性质

$$\|v^{k+1} - v^*\|_H^2 \le \|v^k - v^*\|_H^2 - \|v^k - \tilde{v}^k\|_G^2,\ \ \forall\, v^* \in \mathcal{V}^*. \qquad (7.5)$$

证明. 用 w^* 替代 (7.1) 中任意的 $w \in \Omega$ 并利用单调性, 就得到定理的结论. □

因为 G 正定, 序列 $\{v^k\}$ 是在 H-模下收缩的, (7.5) 是方法收敛的关键不等式. 此外, 利用 (7.1), 马上得到 $O(1/t)$ 收敛速率的结论.

Theorem 7.2. 对变分不等式问题 (4.3), 算法 (6.1) 在条件 (6.2) 满足的时候有遍历意义下的 $O(1/t)$ 收敛速率, 即对任意正整数 t, 都有

$$\theta(\tilde{u}_t) - \theta(u) + (\tilde{w}_t - w)^T F(w) \le \frac{1}{2(t+1)}\|v - v^0\|_H^2,\ \ \forall w \in \Omega, \qquad (7.6)$$

其中 $\tilde{w}_t = \dfrac{1}{(t+1)}\Big(\displaystyle\sum_{k=0}^{t} \tilde{w}^k\Big)$ 是预测序列的算术平均.

证明. 首先, 利用 $(w - \tilde{w}^k)^T F(\tilde{w}^k) = (w - \tilde{w}^k)^T F(w)$, 由 (7.1) 可以得到

$$\theta(\tilde{u}^k) - \theta(u) + (\tilde{w}^k - w)^T F(w) + \frac{1}{2}\|v - v^{k+1}\|_H^2 \le \frac{1}{2}\|v - v^k\|_H^2,\ \ \forall w \in \Omega. \qquad (7.7)$$

将上式从 $k = 0, 1, \ldots, t$ 累加, 再利用凸函数的性质 $\theta\big(\frac{1}{t+1}\sum_{k=0}^t \tilde{u}^k\big) \le \frac{1}{t+1}\big(\sum_{k=0}^t \theta(\tilde{u}^k)\big)$, 就得到 (7.6). 定理得证. □

8 结论与体会

讲述这些算法, 只用到微积分中的梯度, 和线性代数中的矩阵基本运算的知识. 引理 1.1 是凸优化的一阶最优性条件. 用鞍点, 无非是将原问题的解和对偶解一起考虑. 现实生活中, 一个极小总和一个极大联系在一起. 一些基本想法, 还真跟脚在泥土里浸过几年很有关系. 大学数学与日常生活的理念相辅相成, 是本文想说的道理.

爱美之心, 人皆有之. 数学之美, 不是纯数学的专利, 应用与计算也不一定就那么枯燥无味. 用了 §6 的算法框架和收敛性条件, §7 用一页左右的篇幅就给出了收缩性和收敛速率的证明. 不管研究档次高与低, 都力求找到简单统一的规律. 简单, 有人才会看懂使用; 统一, 自己才有美的享受.

References

[1] S. Boyd, N. Parikh, E. Chu, B. Peleato and J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, *Foun. Trends Mach. Learn.*, **3**, 1-122, 2010.

[2] C. H. Chen, B. S. He, Y. Y. Ye and X. M. Yuan, *The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent*, to appear in *Mathematical Programming, Series A*.

[3] E. Esser, M. Möller, S. Osher, G. Sapiro and J. Xin, A convex model for non-negative matrix factorization and dimensionality reduction on physical space, *IEEE Trans. Imag. Process.*, **21**(7), 3239-3252, 2012.

[4] D. Gabay, Applications of the method of multipliers to variational inequalities, *Augmented Lagrange Methods: Applications to the Solution of Boundary-valued Problems*, edited by M. Fortin and R. Glowinski, North Holland, Amsterdam, The Netherlands, pp. 299–331, 1983.

[5] R. Glowinski, *Numerical Methods for Nonlinear Variational Problems*, Springer-Verlag, New York, Berlin, Heidelberg, Tokyo, 1984.

[6] B. S. He, H. Liu, Z.R. Wang and X.M. Yuan, A strictly contractive Peaceman-Rachford splitting method for convex programming, *SIAM Journal on Optimization* **24**, 1011-1040, 2014.

[7] B. S. He, M. Tao and X.M. Yuan, Alternating direction method with Gaussian back substitution for separable convex programming, *SIAM Journal on Optimization* **22**, 313-340, 2012.

[8] B.S. He, M. Tao and X.M. Yuan, A splitting method for separable convex programming, *IMA Journal of Numerical Analysis*, **31**, 394-426, 2015.

[9] B. S. He, M. H. Xu, and X. M. Yuan, Solving large-scale least squares covariance matrix problems by alternating direction methods, *SIAM Journal on Matrix Analysis and Applications* **32**(2011), 136-152.

[10] B. S. He and X. M. Yuan, On the $O(1/n)$ convergence rate of the alternating direction method, *SIAM J. Numerical Analysis* **50**, 700-709, 2012.

[11] B. S. He and X. M. Yuan, On non-ergodic convergence rate of Douglas-Rachford alternating directions method of multipliers, *Numerische Mathematik*, online published.

[12] 何炳生, 凸优化的一阶分裂算法—变分不等式为工具的统一框架, 见 http://math.nju.edu.cn/~hebma 中的《My Talk》

[13] 何炳生, 凸优化和单调变分不等式的收缩算法, 见 http://math.nju.edu.cn/~hebma 中的系列讲义.