说说我的主要研究兴趣(续)

近十年我们在ADMM方面的主要工作

何　炳　生

南方科技大学数学系　　南京大学数学系

十年多以前, 我写了篇《说说我的主要研究兴趣–兼谈华罗庚推广优选法对我的影响》, 以十个为什么的形式, 讲述我选择研究单调变分不等式算法的原因和怎么做这方面的研究. 到那时为止, 我主要关心管理科学中的变分不等式问题, 对只用函数值的方法感兴趣, 致力于研究少用函数值的方法. 其中提到了我对交替方向法的偏爱, 也对自己只能从管理学科粗略地举些优化应用的例子而不甚满意.

2008年, 信息科学中出现的一些优化问题, 让我觉得交替方向法在解决工程技术问题上很快会大有用武之地. 这时, 从我这里硕士毕业的袁晓明, 已经有了七年博士和博士后的历练, 他对交替方向法的浓厚兴趣, 又让我们走到一起. 有了袁的合作, 我们取得国际认可的成果才逐步多了起来.

国内优化界最早接触交替方向法的我, 耳顺以后不知不觉又和学生们一起干了10年. 近来得知有素不相识的青年学者介绍我的工作和主页, 使我觉得该为前面提及的那篇《说说》写个续篇, 告诉关注我工作的学者, 这十年我与学生和朋友主要做了哪些, 同样说说为什么做这些工作的理由.

做研究, 重在享受过程. 我们只是兴趣使然, 做了些量力而行的实际工作. 我这个"老高三"出身的77级本科生, 错过了接受高等教育的黄金年龄, 又与扬长避短相悖, 念了只适合年轻人念的数学. 能做出自成体系的工作并被他人拿来解决不少问题, 这可遇而不可求, 不能不感谢运气.

1　引言和基础知识

我从1997年就开始关注和研究交替方向法, 最近十年, 才考虑用ADMM求解典型的线性约束凸优化问题. 这一节, 我们介绍交替方向类算法要解决的典型问题, 这类算法需要用到的一般的优化原理和普通的大学数学, 同时解释结构型可分离凸优化用交替方向法求解有哪些优点.

1.1　一些典型的优化问题

设 $\mathcal{U}\subset\Re^n$ 是闭凸集, $\theta(u)$ 是 $\Re^n\to\Re$ 的凸函数(不一定可微), 矩阵 $\mathcal{A}\in\Re^{m\times n}$, 向量 $b\in\Re^m$. 这里所说的等式约束凸优化问题的一般形式是

$$\min\{\theta(u)\mid \mathcal{A}u=b,\ u\in\mathcal{U}\}. \tag{1.1}$$

交替方向法 [4, 6, 7, 8] 引起重视, 是问题 (1.1) 往往具有以下的可分离结构: $\theta(u)=\theta_1(x)+\theta_2(y)$, $\mathcal{A}=(A,B)$ 和 $\mathcal{U}=\mathcal{X}\times\mathcal{Y}$, 写起来就是

$$\min\{\theta_1(x)+\theta_2(y)\mid Ax+By=b,\, x\in\mathcal{X},y\in\mathcal{Y}\}. \tag{1.2}$$

利用交替方向法求解 (1.2), 优点是可以利用问题的可分离结构.

三个可分离算子的凸优化问题

$$\min\{\theta_1(x)+\theta_2(y)+\theta_3(z)|Ax+By+Cz=b,\ x\in\mathcal{X},y\in\mathcal{Y},z\in\mathcal{Z}\}, \tag{1.3}$$

相当于在 (1.1) 中置 $\theta(u)=\theta_1(x)+\theta_2(y)+\theta_3(z)$, $\mathcal{A}=(A,B,C)$ 和 $\mathcal{U}=\mathcal{X}\times\mathcal{Y}\times\mathcal{Z}$. 对两个算子的问题有效的方法, 推广到三个算子的问题, 不一定收敛 [3].

1.2 一般的优化原理和变分不等式表述

我们总在变分不等式框架下考虑问题. 因此, 先要用一般的优化原理, 把线性约束的凸优化问题转换成等价的变分不等式问题. 首先, 问题 (1.1) 的拉格朗日函数是定义在 $\mathcal{U} \times \Re^m$ 上的

$$L(u,\lambda) = \theta(u) - \lambda^T(\mathcal{A}u - b). \tag{1.4}$$

如果一对 $w^* = (u^*,\lambda^*) \in \mathcal{U} \times \Re^m$ 满足

$$L_{\lambda \in \Re^m}(u^*,\lambda) \le L(u^*,\lambda^*) \le L_{u\in\mathcal{U}}(u,\lambda^*), \tag{1.5}$$

就称它为 Lagrange 函数的鞍点. 求得了鞍点$w^* = (u^*,\lambda^*)$, 其中的u^* 就是 (1.1) 的解, λ^* 就是相应的最优 Lagrange 乘子, 这就是我们所需要的优化的一般原理. 定义鞍点的联式不等式 (1.5) 分写开来是

$$\begin{cases} u^* \in \mathcal{U}, & L(u,\lambda^*) - L(u^*,\lambda^*) \ge 0, \quad \forall\, u \in \mathcal{U}, \\ \lambda^* \in \Re^m, & L(u^*,\lambda^*) - L(u^*,\lambda) \ge 0, \quad \forall\, \lambda \in \Re^m. \end{cases}$$

根据对 $L(u,\lambda)$ 的具体形式, 以上的鞍点不等式关系可以等价地表示成是下面的变分不等式:

$$\begin{cases} u^* \in \mathcal{U}, & \theta(u) - \theta(u^*) + (u-u^*)^T(-\mathcal{A}^T\lambda^*) \ge 0, \quad \forall\, u \in \mathcal{U}, & \text{(1.6a)} \\ \lambda^* \in \Re^m, & (\lambda-\lambda^*)^T(\mathcal{A}u^* - b) \ge 0, \quad \forall\, \lambda \in \Re^m. & \text{(1.6b)} \end{cases}$$

我们再将 (1.6) 写成紧凑的形式

$$w^* \in \Omega, \quad \theta(u) - \theta(u^*) + (w-w^*)^T F(w^*) \ge 0, \quad \forall\, w \in \Omega, \tag{1.7a}$$

其中

$$w = \begin{pmatrix} u \\ \lambda \end{pmatrix}, \quad F(w) = \begin{pmatrix} -\mathcal{A}^T\lambda \\ \mathcal{A}u - b \end{pmatrix}, \quad \Omega = \mathcal{U} \times \Re^m. \tag{1.7b}$$

从 (1.7) 到 (1.6), 只要对其中任意的 $w \in \Omega$ 分别取 $w = (u,\lambda^*)$ 和 $w = (u^*,\lambda)$ 就得到 (1.6a) 和 (1.6b). 变分不等式 (1.7) 的解集, 用 Ω^* 表示. 以变分不等式 (1.7) 为工具研究凸优化算法, 就像用导函数求二次凸函数的极值, 会带来很大的方便. 这种观点, 正得到越来越多的认可.

1.3 普通的大学数学和增广拉格朗日乘子法

我们需要的普通的大学数学, 是指利用凸函数的性质和一般的微积分知识, 证明下面的引理.

引理 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

$$x^* = \arg\min\{\theta(x) + f(x)\,|\, x \in \mathcal{X}\} \tag{1.8a}$$

的充分必要条件是

$$x^* \in \mathcal{X}, \quad \theta(x) - \theta(x^*) + (x-x^*)^T \nabla f(x^*) \ge 0, \quad \forall\, x \in \mathcal{X}. \tag{1.8b}$$

这个证明相当初等, 也可以在我主页所列报告 [30] 的开头部分找到. 凸优化问题的ADMM类算法的收敛性证明, 基本上就利用变分不等式的表述和这个引理.

优化问题(1.1)的增广拉格朗日函数是在它的拉格朗日函数(1.4)的后面再加一项 $\frac{\beta}{2}\|\mathcal{A}u-b\|^2$, 其中 $\beta>0$ 是给定(但对算法效果还相当敏感)的常数, 记

$$\mathcal{L}_\beta(u,\lambda)=\theta(u)-\lambda^T(\mathcal{A}u-b)+\frac{\beta}{2}\|\mathcal{A}u-b\|^2. \tag{1.9}$$

求解问题 (1.1) 的增广拉格朗日乘子法(ALM) [20, 24] 的 k-步迭代是从给定的 λ^k 开始, 通过

$$\text{(ALM)}\quad \begin{cases} u^{k+1}=\arg\min\left\{\mathcal{L}_\beta(u,\lambda^k)|u\in\mathcal{U}\right\}, & (1.10\text{a})\\ \lambda^{k+1}=\lambda^k-\beta(\mathcal{A}u^{k+1}-b), & (1.10\text{b})\end{cases}$$

求得新的迭代点 $w^{k+1}=(u^{k+1},\lambda^{k+1})$. 因为迭代只要有 λ^k 就可以开始, 像这样的情况, 我们把 u 说成中间变量, λ 称为核心变量. Nocedal 和 Wright 的专著 [22] 的第十七章对增广拉格朗日乘子法的优点有专门的论述. 当然, 它本是用来求解比 (1.1) 更复杂一些的问题.

子问题(1.10a)的类型就是引理1.1中的(1.8a). 引理1.1告诉我们, 你求解了什么具体问题, 就会得到哪些有用的性质. 反过来, 你需要(1.8b)类似的性质, 就去构造并求解(1.8a)那样的问题.

1.4 为什么提倡用交替方向法求解可分离凸优化问题

我们遇到的凸优化问题常常是有可分离结构的 (1.2) 这种类型, 它的增广拉格朗日函数是

$$\mathcal{L}_\beta(x,y,\lambda)=\theta_1(x)+\theta_2(y)-\lambda^T(Ax+By-b)+\frac{\beta}{2}\|Ax+By-b\|^2. \tag{1.11}$$

如果考虑用增广拉格朗日乘子法求解(1.2), 根据 (1.10), k-步迭代是从给定的 λ^k 开始, 求得

$$\begin{cases} (x^{k+1},\,y^{k+1})=\arg\min\big\{\mathcal{L}_\beta(x,y,\lambda^k)\,|\,x\in\mathcal{X},y\in\mathcal{Y}\big\}, & (1.12\text{a})\\ \lambda^{k+1}=\lambda^k-\beta(Ax^{k+1}+By^{k+1}-b). & (1.12\text{b})\end{cases}$$

利用增广 Lagrange 函数的表达式 (1.11), 子问题 (1.12a) 的形式是

$$(x^{k+1},\,y^{k+1})=\arg\min\left\{\begin{matrix}\theta_1(x)+\theta_2(y)-(\lambda^k)^T(Ax+By-b)\\ +\frac{\beta}{2}\|Ax+By-b\|^2\end{matrix}\;\middle|\;\begin{matrix}x\in\mathcal{X}\\ y\in\mathcal{Y}\end{matrix}\right\}.$$

这种处理方式, 缺点是没有利用问题的可分离结构, 求解这样的子问题有时会无从下手.

针对 (1.12) 中子问题难解的情况, 考虑将 (x,y) 子问题通过松弛分拆开做, 就得到下面的**乘子交替方向法(ADMM)**. 这个方法的 k-步迭代是从给定的核心变量 (y^k,λ^k) 开始, 通过

$$\text{(ADMM)}\quad \begin{cases} x^{k+1}=\arg\min\big\{\mathcal{L}_\beta(x,y^k,\lambda^k)\;\big|\;x\in\mathcal{X}\big\}, & (1.13\text{a})\\ y^{k+1}=\arg\min\big\{\mathcal{L}_\beta(x^{k+1},y,\lambda^k)\;\big|\;y\in\mathcal{Y}\big\}, & (1.13\text{b})\\ \lambda^{k+1}=\lambda^k-\beta(Ax^{k+1}+By^{k+1}-b), & (1.13\text{c})\end{cases}$$

得到 $(x^{k+1},y^{k+1},\lambda^{k+1})$. 换句话说, 乘子交替方向法实际上是处理可分离结构型优化问题 (1.2) 的松弛了的增广拉格朗日乘子法. 子问题 (1.13a) 和 (1.13b) 分别是

$$\min_{x\in\mathcal{X}}\{\theta_1(x)+\frac{\beta}{2}\|Ax+(By^k-b-\frac{1}{\beta}\lambda^k)\|^2\}\quad \text{和}\quad \min_{y\in\mathcal{Y}}\{\theta_2(y)+\frac{\beta}{2}\|By+(Ax^{k+1}-b-\frac{1}{\beta}\lambda^k)\|^2\}$$

这样单个变量的问题. 这里的 β 怎么取, 还是有讲究. 2009 年 10 月上海大学主办的华东地区运筹学与控制论博士论坛安排我做大会报告的时候, 我就以“信息技术中的凸优化问题和交替方向法求解”为题, 呼吁青年学者注意交替方向法. 第二年, 网上就有了 Boyd 他们受到热捧的关于乘子交替方向法的文章 [1].

2 两个算子问题交替方向法方面做了哪些改进

对增广拉格朗日乘子法松弛, 就得到了求解可分离结构问题 (1.2) 的有效方法 ADMM (1.13), 想法相当简单. 利用变分不等式, 证明经典的ADMM产生的迭代序列 $\{v^k\} = \{(y^k, \lambda^k)\}$ 满足

$$\|v^{k+1} - v^*\|_H^2 \le \|v^k - v^*\|_H^2 - \|v^k - v^{k+1}\|_H^2, \quad 其中 \quad H = \begin{pmatrix} \beta B^{\mathrm{T}} B & 0 \\ 0 & \frac{1}{\beta} I \end{pmatrix}. \tag{2.1}$$

这个证明相当容易(见[30]或[31]的第十一讲), 写黑板我只需花不到半个小时. 据此可以直接得到

$$\sum_{k=0}^{\infty} \|v^k - v^{k+1}\|_H^2 \le \|v^0 - v^*\|_H^2. \tag{2.2}$$

这么漂亮又简单的工作让人家做成了. 羡慕之余, 自然要问, 我们还能做点什么？出于这种考虑, 对两个算子的问题交替方向法本身, 我们主要提出了: 1) PPA 意义下延拓的乘子交替方向法; 2) 对称的乘子交替方向法; 和 3) 不定意义下的线性化乘子交替方向法.

2.1 PPA 意义下延拓的乘子交替方向法

PPA 意义下可以延拓的乘子交替方向法发表在[2], 具体做法是 k-步迭代从给定的核心变量 (y^k, λ^k) 开始, 将经典的 ADMM (1.13) 中求解 y-子问题和校正 λ 的顺序交换, 将通过

$$\left\{ \begin{array}{lcl} x^{k+1} & = & \arg\min\{\mathcal{L}_\beta(x, y^k, \lambda^k) \,|\, x \in \mathcal{X}\}, \\ \lambda^{k+1} & = & \lambda^k - \beta(Ax^{k+1} + By^k - b), \\ y^{k+1} & = & \arg\min\{\mathcal{L}_\beta(x^{k+1}, y, \lambda^{k+1}) \,|\, y \in \mathcal{Y}\}, \end{array} \right. \tag{2.3a}$$

得到的 $w^{k+1} = (x^{k+1}, y^{k+1}, \lambda^{k+1})$ 作为预测点, 然后再做松弛延拓

$$\left\{ \begin{array}{lcl} y^{k+1} & := & y^k - \gamma(y^k - y^{k+1}), \\ \lambda^{k+1} & := & \lambda^k - \gamma(\lambda^k - \lambda^{k+1}), \end{array} \right. \tag{2.3b}$$

这里往往取 $\gamma = 1.5 \in (0, 2)$. 赋值号 “:=” 表示 (2.3b) 右端的 (y^{k+1}, λ^{k+1}) 是由算法的前半部分 (2.3a) 产生的. (2.3b) 左端才是为下一步迭代开始所提供的 (y^{k+1}, λ^{k+1}). 对多数问题, 这样做往往能加快收敛.

2.2 对称更新乘子的交替方向法

求解 (1.2) 的经典的乘子交替方向法是 (1.13). 从问题 (1.2) 本身看, 原始变量 x 和 y 是平等的. 在算法设计上平等对待 x 子问题和 y 子问题, 也是最自然不过的考虑. 因此我们采用对称的交替方向法 [13]. 它的 k-步迭代从给定的 (y^k, λ^k) 开始, 通过

$$\left\{ \begin{array}{l} x^{k+1} = \arg\min\{\mathcal{L}_\beta(x, y^k, \lambda^k) \,|\, x \in \mathcal{X}\}, \\ \lambda^{k+\frac{1}{2}} = \lambda^k - \mu\beta(Ax^{k+1} + By^k - b), \\ y^{k+1} = \arg\min\{\mathcal{L}_\beta(x^{k+1}, y, \lambda^{k+\frac{1}{2}}) \,|\, y \in \mathcal{Y}\}, \\ \lambda^{k+1} = \lambda^{k+\frac{1}{2}} - \mu\beta(Ax^{k+1} + By^{k+1} - b). \end{array} \right. \tag{2.4}$$

得到新的迭代点 $w^{k+1} = (x^{k+1}, y^{k+1}, \lambda^{k+1})$, 其中 $\mu \in (0, 1)$ (我们通常取 $\mu = 0.9$). 这样做, 想法合理, 事实上也加快收敛. 要注意的是当 $\mu = 1$ 时, (2.4)就成了Peaceman-Rachford方法[21, 23], 对此, 可以举出不收敛的例子.

2.3 线性化交替方向法上的进展

ADMM 方法 (1.13) 中, 子问题 (1.13a) 和 (1.13b) 的具体形式分别是对已知的p^k和q^k, 求解

$$\min\{\theta_1(x)+\frac{\beta}{2}\|Ax-p^k\|^2\,|\,x\in\mathcal{X}\}\quad \text{和}\quad \min\{\theta_2(y)+\frac{\beta}{2}\|By-q^k\|^2\,|\,y\in\mathcal{Y}\}.$$

在一些实际应用中, 由于矩阵 A 或 B 的结构, 其中会有一个子问题难解. 我们不妨设这个比较难解的问题是 (1.13b), 然后在它的目标函数后面加个特殊的正则项, 把方法 (1.13) 改成

$$\text{(L-ADMM)}\quad \begin{cases} x^{k+1}=\arg\min\{\mathcal{L}_\beta(x,y^k,\lambda^k)\;\big|\;x\in\mathcal{X}\}, & (2.5a)\\ y^{k+1}=\arg\min\{\mathcal{L}_\beta(x^{k+1},y,\lambda^k)+\frac{1}{2}\|y-y^k\|_{D_B}^2\;\big|\;y\in\mathcal{Y}\}, & (2.5b)\\ \lambda^{k+1}=\lambda^k-\beta(Ax^{k+1}+By^{k+1}-b), & (2.5c)\end{cases}$$

其中

$$D_B=sI-\beta B^TB. \tag{2.6}$$

由于改变目标函数中的常数项不影响解, 因此 (2.5b) 就变成了形如

$$y^{k+1}=\arg\min\{\theta_2(y)+\frac{s}{2}\big\|y-\big[y^k-\frac{1}{s}B^T[\beta(Ax^{k+1}+By^k-b)-\lambda^k]\big]\big\|^2\;\big|\;y\in\mathcal{Y}\}$$

这样简单一些的子问题. 这常常被说成是“线性化”处理. 实际上是

$$\text{用}\quad \frac{s}{2}\|y-y^k\|^2\quad \text{去代替}\quad \frac{\beta}{2}\|B(y-y^k)\|^2.$$

为了理论上保证收敛, 对于固定(不能随意变小)的 β, 人们要求 (2.6) 中的参数 $s\ge\beta\|B^TB\|$ [26, 27]. 大家同时知道, 过大的 $s>0$, 会影响收敛速度.

2017 年, 我们在这方面有了大的进展. 在 [14] 中, 我们证明了可以缩小这个参数 s. 只要 $s>\frac{3}{4}\beta\|B^TB\|$ 方法 (2.5) 就可以保证收敛. 当 $s<\frac{3}{4}\beta\|B^TB\|$ 就有不收敛的例子. 换句话说, 在这一方面, 我们做到了最优, 我们曾经为此自豪.

3 对三个算子问题提出了哪些交替方向类方法

求解三个可分离算子问题(1.3)的ADMM类算法, 以前没人讨论过. 我们记问题 (1.3) 的增广拉格朗日函数为(与两个算子的符号有区别)

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

采用直接推广的乘子交替方向法, k 步迭代是从给定的核心变量 $v^k=(y^k,z^k,\lambda^k)$ 出发, 通过

$$\begin{cases} x^{k+1} &=& \arg\min\{\mathcal{L}_\beta^3(x,y^k,z^k,\lambda^k)\;\big|\;x\in\mathcal{X}\},\\ y^{k+1} &=& \arg\min\{\mathcal{L}_\beta^3(x^{k+1},y,z^k,\lambda^k)\;\big|\;y\in\mathcal{Y}\},\\ z^{k+1} &=& \arg\min\{\mathcal{L}_\beta^3(x^{k+1},y^{k+1},z,\lambda^k)\;\big|\;z\in\mathcal{Z}\},\\ \lambda^{k+1} &=& \lambda^k-\beta(Ax^{k+1}+By^{k+1}+Cz^{k+1}-b)\end{cases} \tag{3.1}$$

求得新的迭代点 $w^{k+1}=(x^{k+1},y^{k+1},z^{k+1},\lambda^{k+1})$. 在 [3] 中, 我们证明了当问题 (1.3) 中的矩阵 A, B, C 的列向量空间有两个相互正交的时候, 方法 (3.1) 是收敛的, 因为这时它相当于两个算子的问题. 在那篇文章中, 我们也给出了用 (3.1) 求解三个算子问题不收敛的例子.

在不清楚直接推广的 ADMM (3.1) 是否收敛之前, 我们就着手对三个算子的问题提出一些修正的ADMM类算法. 修正的原则是尽量少做改动, 保持 ADMM 好品性. 对问题不加任何额外条件, 对经典 ADMM 中需要调比选取的 β, 不做任何限制. 一句话, 只对方法本身动点小手术!

3.1 部分平行分裂 ALM 的预测校正方法

为了更接近 (3.1), 还是把 x 当成中间变量, 迭代从 $v^k=(y^k,z^k,\lambda^k)$ 到 $v^{k+1}=(y^{k+1},z^{k+1},\lambda^{k+1})$, 平行处理 y, z-子问题再更新 λ. 具体说来, 就是通过

$$\left\{\begin{array}{ll}
x^{k+1}=\arg\min\left\{\mathcal{L}_\beta^3(x,y^k,z^k,\lambda^k)\;\middle|\;x\in\mathcal{X}\right\}, & \text{(3.2a)}\\
y^{k+1}=\arg\min\left\{\mathcal{L}_\beta^3(x^{k+1},y,z^k,\lambda^k)\;\middle|\;y\in\mathcal{Y}\right\}, & \text{(3.2b)}\\
z^{k+1}=\arg\min\left\{\mathcal{L}_\beta^3(x^{k+1},y^k,z,\lambda^k)\;\middle|\;z\in\mathcal{Z}\right\}, & \text{(3.2c)}\\
\lambda^{k+1}=\lambda^k-\beta(Ax^{k+1}+By^{k+1}+Cz^{k+1}-b), & \text{(3.2d)}
\end{array}\right.$$

生成的 $(x^{k+1},y^{k+1},z^{k+1},\lambda^{k+1})$ 当成预测点. y, z 子问题 (3.2b) 和 (3.2c) 平行了, 太自由, 包括据此更新的 λ, 都需要校正. 校正公式是

$$v^{k+1}:=v^k-\alpha(v^k-v^{k+1}),\quad \alpha\in(0,2-\sqrt{2}). \tag{3.3}$$

譬如说, 我们可以取 $\alpha=0.55$. 注意到 (3.3) 右端的 $v^{k+1}=(y^{k+1},z^{k+1},\lambda^{k+1})$ 是由(3.2) 提供的. 换句话说, 这里的校正就是把走的太“远”的 v^{k+1} 往回拉一点. 预测-校正, 是我们从投影收缩算法开始的算法框架. 预测只是提供收缩方向, 校正时计算步长, 可以确保在正定的 H-模下, $\|v^k-v^*\|_H$ 下降收缩. 方法 2009 年发表在 [10], 系列讲义 [31] 的第十五讲中也有详细论述.

3.2 带高斯回代的 ADMM 方法

直接推广的乘子交替方向法 (3.1) 对三个算子的问题 (1.3) 不能保证收敛, 是因为它们处理有关核心变量的 y 和 z-子问题不公平. 采取补救的办法是将 (3.1) 提供的 $(y^{k+1},z^{k+1},\lambda^{k+1})$ 当成预测点, 由于为下一步迭代只需要准备 $(By^{k+1},Cz^{k+1},\lambda^{k+1})$, 校正公式为

$$\begin{pmatrix}By^{k+1}\\Cz^{k+1}\\\lambda^{k+1}\end{pmatrix}:=\begin{pmatrix}By^{k}\\Cz^{k}\\\lambda^{k}\end{pmatrix}-\nu\begin{pmatrix}I&-I&0\\0&I&0\\0&0&I\end{pmatrix}\begin{pmatrix}B(y^k-y^{k+1})\\C(z^k-z^{k+1})\\\lambda^k-\lambda^{k+1}\end{pmatrix}. \tag{3.4}$$

其中 $\nu\in(0,1)$, 右端的 $(y^{k+1},z^{k+1},\lambda^{k+1})$ 是由 (3.1) 提供的. 这个方法发表在 [15]. 想法是 (3.1) 处理 y 和 z 子问题有先后, 不公平, 就要做找补, 调整. 事实上, 也可以只校正 y 和 z (无需校正λ). 也就是说, 做比 (3.4) 更简单的

$$\begin{pmatrix}By^{k+1}\\Cz^{k+1}\end{pmatrix}:=\begin{pmatrix}By^{k}\\Cz^{k}\end{pmatrix}-\nu\begin{pmatrix}I&-I\\0&I\end{pmatrix}\begin{pmatrix}By^k-By^{k+1}\\Cz^k-Cz^{k+1}\end{pmatrix}. \tag{3.5}$$

收敛性的简单证明可以从我主页的报告序列 [30] 中找到. 这个处理三个算子的方法, 在直接推广的ADMM (3.1) 类算法的基础上做的修改是微乎其微的.

3.3 部分平行并加正则项的 ADMM 方法

§3.2 中的方法将 (3.1) 的输出作为预测点校正, 是因为 (3.1) 处理 y 和 z 子问题有先后, 不公平. 这里我们考虑强制公平, 平行求解 y, z-子问题, 不做后处理, 而是给这两个子问题预先都加个正则项. 方法写起来就是

$$\begin{cases} x^{k+1} = \arg\min\left\{\mathcal{L}_\beta^3(x, y^k, z^k, \lambda^k) \,\middle|\, x \in \mathcal{X}\right\}, \\ y^{k+1} = \arg\min\left\{\mathcal{L}_\beta^3(x^{k+1}, y, z^k, \lambda^k) + \frac{\tau}{2}\beta\|B(y-y^k)\|^2 \middle| y \in \mathcal{Y}\right\}, \\ z^{k+1} = \arg\min\left\{\mathcal{L}_\beta^3(x^{k+1}, y^k, z, \lambda^k) + \frac{\tau}{2}\beta\|C(z-z^k)\|^2 \middle| z \in \mathcal{Z}\right\}, \\ \lambda^{k+1} = \lambda^k - \beta(Ax^{k+1} + By^{k+1} + Cz^{k+1} - b), \end{cases} \tag{3.6}$$

其中 $\tau > 1$. 上述做法相当于

$$\begin{cases} x^{k+1} = \arg\min\{\theta_1(x) + \frac{\beta}{2}\|Ax + By^k + Cz^k - b - \frac{1}{\beta}\lambda^k\|^2 \,|\, x \in \mathcal{X}\}, \\ \qquad \lambda^{k+\frac{1}{2}} = \lambda^k - \beta(Ax^{k+1} + By^k + Cz^k - b) \\ y^{k+1} = \arg\min\{\theta_2(y) - (\lambda^{k+\frac{1}{2}})^T By + \frac{\mu\beta}{2}\|B(y-y^k)\|^2 \,|\, y \in \mathcal{Y}\}, \\ z^{k+1} = \arg\min\{\theta_3(z) - (\lambda^{k+\frac{1}{2}})^T Cz + \frac{\mu\beta}{2}\|C(z-z^k)\|^2 \,|\, z \in \mathcal{Z}\}, \\ \lambda^{k+1} = \lambda^k - \beta(Ax^{k+1} + By^{k+1} + Cz^{k+1} - b), \end{cases} \tag{3.7}$$

其中 $\mu = \tau + 1$. 这类发表在 [16, 25] 的算法思想是: 让 y 和 z 太自由, 以后又不准备校正, 那就用加正则项让它们不会走得太远. [16] 中的方法 (3.7) 被 UCLA Osher 教授的课题组成功用来求解图像降维问题[5]. 我们要求 $\mu > 2$, 他们因此取 $\mu = 2.01$, 应该是这个参数取大了不好.

> 2017 年, 我们在这方面有了大的进展. 在 [19] 中, 我们证明了可以缩小 (3.6) 中的参数 τ, 满足 $\tau > \frac{1}{2}$, 但对 $\tau < \frac{1}{2}$ 给出了不收敛的例子. 这相当于在 (3.7) 中, 只要取 $\mu > 1.5$ 就可以了. 如果我们早有这个结果, Osher 课题组在[5]中或许就取 $\mu = 1.51$ 了.

4 凸优化分裂收缩算法的统一框架

这些年的一个主要成果是对求解单调变分不等式 (1.7) 的算法建立了一个统一框架. 把算法都看成预测–校正方法[11, 30, 29], 每步迭代都分拆成预测和校正.

> **[预测]** 算法的 k-步迭代从 v^k 开始, 通过求解一些子问题, 产生的预测点 $\tilde{w}^k \in \Omega$, 使得
>
> $$\theta(u) - \theta(\tilde{u}^k) + (w - \tilde{w}^k)^T F(\tilde{w}^k) \geq (v - \tilde{v}^k)^T Q(v^k - \tilde{v}^k), \ \forall\, w \in \Omega, \tag{4.1}$$
>
> 成立. 其中 $Q^T + Q$ 正定.

> **[校正]** 我们用校正公式
>
> $$v^{k+1} = v^k - M(v^k - \tilde{v}^k), \tag{4.2}$$
>
> 产生新的迭代点.

前面提到的算法, 都可以归进这个框架. 一些新的算法, 细细分析起来, 往往不是在预测矩阵 Q 上加点“酱”, 就得在校正矩阵 M 上加点“醋”. 对预测-校正方法 (4.1) – (4.2), 有下面的特别容易检测的保证收敛的充分条件:

收敛性条件 如果预测-校正公式中的矩阵 Q 和M 满足

$$H = QM^{-1} \succ 0, \tag{4.3a}$$

和

$$G = Q^T + Q - M^T HM \succ 0, \tag{4.3b}$$

迭代序列就会收敛到一个解点, 并且具有 $O(1/t)$ 的收敛速率.

需要强调的是, 算法中不需要知道 H, 也不需要 M^{-1} 的信息. 对统一框架中的预测-校正算法, 验证上述条件满足了, 收敛性的一些理想性质就都有了.

4.1 统一框架下的收缩性质

对统一框架的算法, 我们证明了以下的一些收敛性质.

定理 4.1. 设 $\{\tilde{w}^k\}$, $\{v^k\}$ 是求解变分不等式 (1.7) 的预测-校正方法 (4.1)-(4.2) 生成的序列. 如果条件 (4.3) 成立, 那么有

$$\begin{aligned}&\theta(u) - \theta(\tilde{u}^k) + (w - \tilde{w}^k)^T F(\tilde{w}^k) \\ &\quad \ge \frac{1}{2}\big(\|v - v^{k+1}\|_H^2 - \|v - v^k\|_H^2\big) + \frac{1}{2}\|v^k - \tilde{v}^k\|_G^2, \ \ \forall w \in \Omega.\end{aligned} \tag{4.4}$$

我常常挂在嘴边的“中学的数理基础”, 是指上述定理证明主要依靠下面的初等恒等式

$$(a-b)^T H(c-d) = \frac{1}{2}\Big\{\|a-d\|_H^2 - \|a-c\|_H^2\Big\} + \frac{1}{2}\Big\{\|c-b\|_H^2 - \|d-b\|_H^2\Big\}.$$

将 (4.4)中任意的 $w \in \Omega$ 设为 w^*, 利用单调性, 得到迭代序列收缩性定理特别简单.

定理 4.2. 设 $\{\tilde{w}^k\}$, $\{v^k\}$ 是求解变分不等式 (1.7) 的预测-校正方法 (4.1)-(4.2) 生成的序列. 如果条件 (4.3) 成立, 那么有

$$\|v^{k+1} - v^*\|_H^2 \le \|v^k - v^*\|_H^2 - \|v^k - \tilde{v}^k\|_G^2, \quad \forall v^* \in \mathcal{V}^*. \tag{4.5}$$

4.2 迭代复杂性

统一框架让我们津津乐道的, 是利用它证明遍历意义下的和点列意义下的迭代复杂性就变得相当容易. 对经典的ADMM (1.13) 产生的迭代序列, 通过定义

$$\tilde{x}^k = x^{k+1}, \qquad \tilde{y}^k = y^{k+1}, \qquad \tilde{\lambda}^k = \lambda^k - \beta(Ax^{k+1} + By^k - b)$$

利用统一框架, 我们在文 [17] 中证明了遍历意义下迭代复杂性:

定理 4.3. 设 $\{\tilde{w}^k\}$, $\{v^k\}$ 是求解变分不等式 (1.7) 的预测-校正方法 (4.1)-(4.2) 生成的序列, 条件 (4.3) (此时G只要求半正定) 成立. 记

$$\tilde{w}_t = \frac{1}{t+1}\sum_{k=0}^{t} \tilde{w}^k. \tag{4.6}$$

那么, 对任意的正整数 $t > 0$, $\tilde{w}_t \in \Omega$ 并

$$\theta(\tilde{u}_t) - \theta(u) + (\tilde{w}_t - w)^T F(w) \le \frac{1}{2(t+1)}\|v - v^0\|_H^2, \quad \forall w \in \Omega. \tag{4.7}$$

此外, 我们在文 [18] 中证明了交替方向法产生的序列 $\{\|v^k - v^{k+1}\|_H\}$ 的单调性.

定理 4.4. 设 $\{v^k\}$ 是用交替方向法 (1.13) 求解 (1.2) 得到的序列, 那么有

$$\|v^k - v^{k+1}\|_H^2 \le \|v^{k-1} - v^k\|_H^2. \tag{4.8}$$

由 (2.2) 和 (4.8) 得到, 对任意的正整数 $t > 0$,

$$\|v^t - v^{t+1}\|_H^2 \le \frac{1}{t+1}\Big(\sum_{k=0}^{t}\|v^k - v^{k+1}\|_H^2\Big) \le \frac{1}{t+1}\|v^0 - v^*\|_H^2. \tag{4.9}$$

不等式(4.7)和(4.9)的左端用来检测误差, 右端的量都以迭代次数做分母, 这是我们就得到迭代复杂性结论的主要依据.

5 一点感慨和体会

本文叙述的最近十年我和学生们及朋友在交替方向法方面的一些主要工作. 除了两个最新的成果[14, 19], 其他都在我的综述文章《我和乘子交替方向法20年》[29]中用统一框架给出了证明.

越来越多的学者注意到, 我(和我倡导)的研究工作, 应该算自成一个体系. 从变分不等式的投影收缩算法到以ADMM为代表的凸优化的分裂收缩算法[9, 12, 28], 一个模式, 一条主线. 其中的主要算法, 都曾被他人用来成功解决问题. 方法要能被人采用, 简单明了是必须的.

交替方向法近年这么火热, 我们的工作得到许多名家的认可, 这是 20 年前我开始做交替方向法没有想过的. 如果说我在学术研究的道路上捡到了几个贝壳, 那么一点经验就是做学术研究无需跟风. 不要让风言风语动摇了自己的专注, 因为它夺不走我们成功的愉悦.

我考上高中报到后的第一课, 老师告知我们42个同班同学中有八个在初中当过班长(没有我). 从那个时候开始, 我就知道, 这个世界, 不缺领军人物. 能干自己想干的, 不受干扰, 亦蛮好的!

数学研究之余, 我也读些闲书. 只是年岁大了, 记不住几句. 偶尔读到哲人们用他们的语言, 表达了我多年想表达而没有表达出(或者没有能力表达清)的意思, 也会拍手叫绝. 哲学家周国平说: “真正的使命感无非是对自己选定并且正从事的工作的一种热爱罢了. 遇见这样的人, 我的血缘本能就会把他们认做我的亲兄弟”. 这几句我觉得很有道理.

References

[1] Boyd S, Parikh N, Chu E, Peleato B and Eckstein J. Distributed optimization and statistical learning via the alternating direction method of multipliers, *Foundations and Trends in Machine Learning*, 2011, 3: 1–122.

[2] Cai X J, Gu G Y, He B S and Yuan X M. A proximal point algorithms revisit on the alternating direction method of multipliers, *Science China Mathematics*, 2013, 56: 2179–2186.

[3] Chen C H, He B S, Ye Y Y and Yuan X M. The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent, *Mathematical Programming*, 2016, 155: 57–79.

[4] Douglas J and Rachford H H. On the numerical solution of the heat conduction problem in 2 and 3 space variables, *Trans. Amer. Math. Soc.*, 1956, 82: 421 – 439.

[5] Esser E, Möller M, Osher S, et al. A convex model for non-negative matrix factorization and dimensionality reduction on physical space [J]. *Transactions on Image Processing*, 2012, 21: 3239-3252.

[6] Gabay D and Mercier M. A dual algorithm for the solution of nonlinear variational problems via finite-element approximations, Comput. Math. Appl., 1976, 2: 17 – 40.

[7] Glowinski R. *Numerical methods for nonlinear variational problems*, Springer-Verlag, New York, Berlin, Heidelberg, Tokyo, 1984.

[8] Glowinski R and Marrocco A. Approximation parélḿents finis dórdre un et résolution par pénalisation-dualité d'une classe de problémes non linéaires, RAIRO, R2 1975, 41 – 76.

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

[10] He B S. Parallel splitting augmented Lagrangian methods for monotone structured variational inequalities, *Computational Optimization and Applications*, 2009, 42: 195–212.

[11] He B S. PPA-Like contraction methods for convex optimization: A framework using variational inequality approach, *J. Oper. Res. Soc. China*, 2015, 3: 391–420.

[12] He B S, Liao L Z, Han D R and Yang H. A new inexact alternating directions method for monotone variational inequalities, *Mathematical Programming*, 2002, 92: 103–118.

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

[14] He B S, Ma F, Yuan X M. Optimal Linearized Alternating Direction Method of Multipliers for Convex Programming, http://www.optimization-online.org/DB_HTML/2017/09/6228.html

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

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

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

[18] He B S and Yuan X M. On non-ergodic convergence rate of Douglas-Rachford alternating directions method of multipliers, *Numerische Mathematik*, 2015, 130: 567–577.

[19] He B S and Yuan X M. On the optimal proximal parameter of an ADMM-like splitting method for separable convex programming, http://www.optimization-online.org/DB_HTML/2017/10/6235.html

[20] Hestenes M R. Multiplier and gradient methods, *Journal of Optimization Theory and Applications*, 1969, 4: 302–320.

[21] Lions P. L and Mercier B. Splitting algorithms for the sum of two nonlinear operators, SIAM J. Numer. Anal., 1979, 16: 964 – 979.

[22] Nocedal J and Wright S. *Numerical optimization*, Springer Verlag, New York, 1999.

[23] Peaceman D. W and Rachford H. H. The numerical solution of parabolic and elliptic differential equations, J. Soc. Indust. Appl. Math., 1955, 3: 28 – 41.

[24] Powell M. A method for nonlinear constraints in minimization problems, in Optimization, R. Fletcher, ed., Academic Press, New York, NY, 1969, 283–298.

[25] Tao M and Yuan X M. Recovering low-rank and sparse components of matrices from incomplete and noisy observations *SIAM Journal on Optimization*, 2011, 21: 57-81.

[26] Yang J F and Yuan X M. Linearized augmented Lagrangian and alternating direction methods for nuclear norm minimization *Mathematics of Computation*, 2013, 82: 301-329

[27] Zhang X. Q, Burger M and Osher S. A unified primal-dual algorithm framework based on Bregman iteration, J. Sci. Comput., 2010, 46: 20 – 46.

[28] 何炳生. 从变分不等式的投影收缩算法到凸优化的分裂收缩算法, 高等学校计算数学学报, 2016, 38: 74–96.

[29] 何炳生. 我和乘子交替方向法20年, 运筹学学报, 2018, 22: 1–31.

[30] 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://maths.nju.edu. cn/~hebma 中的《My Talk》.

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