一种内外迭代方法求解三凸函数和优化问题

唐玉超¹, 吴国荣², 朱传喜^{1*}

1. 南昌大学理学院数学系, 南昌 330031;

2. BRIC and Department of Radiology, University of North Carolina, Chapel Hill, NC 27599, USA

E-mail: hhaao1331@163.com, guorong-wu@med.unc.edu, chuanxizhu@126.com

收稿日期: 2017-05-18; 接受日期: 2018-02-22; 网络出版日期: 2019-04-23; * 通信作者

国家自然科学基金 (批准号: 11661056, 11401293 和 11771198) 资助项目

摘要 近年来, 关于多个凸函数和的优化问题受到广泛关注. 本文研究三个凸函数和 $f(x) + g(x) + h(Bx)$ 的一类凸优化问题, 其中 $f(x)$ 可微且具有 Lipschitz 连续梯度, $g(x)$ 和 $h(x)$ 是正则下半连续简单凸函数, B 是一个有界线性算子. 此类优化问题在信号恢复和图像处理等实际问题中有着广泛的应用. 为充分利用问题中的可微函数, 本文基于向前向后分裂算法和三算子分裂算法框架, 建立若干具有内外迭代形式的算法. 在推导迭代算法的过程中, 本文提出基于对偶和原始对偶方法求解函数 $g + h \circ B$ 和 $h \circ B$ 的邻近算子. 在对参数一定假设条件下, 本文证明所提出的迭代算法收敛性. 通过与 Condat 和 Vu 算法、原始对偶不动点 (primal-dual fixed point, PDFP) 算法和原始对偶三算子 (primal-dual three-operator, PD3O) 算法比较, 建立三种迭代算法与本文提出的迭代算法之间的联系. 最后, 通过对融合 Lasso 问题、约束全变分正则化问题和低秩全变分图像超分辨率重建问题实施一系列数值实验, 验证所提出的迭代算法的有效性.

关键词 向前向后分裂算法 三算子分裂算法 对偶 原始对偶 全变分

MSC (2010) 主题分类 90C25, 65K10

1 引言

信号和图像处理、机器学习和统计预测等许多问题都可以归结为求解两个及以上凸函数和的优化问题, 在过去的十多年里, 许多有效的迭代算法被提出. 这里简要回顾一些具有代表性的迭代算法. 例如, 向前向后分裂算法^[1]被广泛地应用于求解下面的优化问题:

$$\min_{x \in X} f(x) + g(x), \quad (1.1)$$

其中 X 是给定实 Hilbert 空间, $f: X \rightarrow \mathbb{R}$ 是凸可微函数且其梯度满足 Lipschitz 连续性, $g \in \Gamma_0(X)$ 是简单的并且可能是非光滑的, 这里 $\Gamma_0(X)$ 表示定义在 Hilbert 空间 X 到 $(-\infty, +\infty]$ 的所有正则的

英文引用格式: Tang Y C, Wu G R, Zhu C X. An inner-outer iteration method for solving convex optimization problems involving the sum of three convex functions (in Chinese). Sci Sin Math, 2019, 49: 831–858, doi: 10.1360/SCM-2017-0313

下半连续凸函数全体. 当 $g(x) = \lambda\|x\|_1$ 且 $f(x) = \frac{1}{2}\|Ax - b\|_2^2$ 时, 其中 $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^n$, $\lambda > 0$, 向前向后分裂算法等价于迭代收缩阈值算法 (iterative shrinkage-thresholding algorithm, ISTA) [2]. 作为迭代收缩阈值算法的推广, Beck 和 Teboulle [3] 在 Nestorv 加速梯度法 [4] 的基础上提出了一种快速迭代收缩阈值算法 (fast iterative shrinkage-thresholding algorithm, FISTA). 当 $g(x)$ 为某些非凸函数时, 例如, 当 ℓ_p ($0 < p < 1$) 拟范数时, 相应向前向后分裂算法的收敛性研究可参见文献 [5, 6] 等. 基于 ℓ_1 范数等的稀疏优化模型在压缩感知理论 [7] 中得到广泛研究和应用. 对于二维信号 (如图像), 全变分 (total variation, TV) [8] 由于其具有保持分片常数图像边缘的能力而被广泛应用于图像恢复和图像重建等问题 (如文献 [9–14]). Beck 和 Teboulle [15] 运用快速迭代收缩阈值算法求解全变分图像去模糊问题. 由于全变分的邻近算子没有封闭形式的显示解, Beck 和 Teboulle [15] 借助 Chambolle [16] 对偶梯度方法思想来解决这个子问题. 注意到, 全变分可以表示为一阶离散梯度差分矩阵和凸函数组合的形式. 为了求解全变分正则化问题, 下面一般的优化问题受到了极大的关注:

$$\min_{x \in X} f(x) + h(Bx), \quad (1.2)$$

其中 X 和 Y 是给定实 Hilbert 空间, $f \in \Gamma_0(X)$, $h \in \Gamma_0(Y)$, $B : X \rightarrow Y$ 是有界线性算子. 在本文中, 总是假定 f 和 h 是简单的, 意味着它们的邻近算子具有显示解. Chambolle 和 Pock [17] 首次提出了一种原始对偶邻近点算法求解问题 (1.2), 这里的对偶是在 Fenchel 共轭的意义下定义的. 文献 [18–20] 分别提出了原始对偶混合梯度方法 (primal-dual hybrid gradient algorithm, PDHG) 求解问题 (1.2), 虽然它与原始对偶邻近点算法的推导角度不同, 但两者基本是同一种算法. He 和 Yuan [21] 指出了原始对偶邻近点算法等价于邻近点算法 (proximal point algorithm, PPA), 从而不仅大大简化了算法的收敛性分析, 而且提供了借助算子分裂算法求解多个凸函数和的优化问题的一种解决思路. 原始对偶邻近点算法属于原始对偶方法的一种, 它在每步迭代中分别更新原始和对偶变量. 关于原始对偶邻近点算法的最新进展, 可以参见文献 [22, 23] 等. 另一种流行的求解问题 (1.2) 的方法是交替方向乘子法 (alternating direction method of multipliers, ADMM) [24]. 交替方向乘子法被证明是等价于 Douglas-Rachford 分裂方法应用于问题 (1.2) 的对偶问题 [25]. 通过将原始对偶邻近点算法与交替方向乘子法比较, 可以得出, 前者只需要算子 B 及其共轭算子的矩阵向量乘法, 而交替方向乘子法需要求解一个非平凡子问题. 如果 f 是可微且具有 Lipschitz 连续梯度, Chen 等 [26] 提出了一种基于邻近不动点算法 [27] 的原始对偶不动点算法 (primal-dual fixed point algorithm based on proximity operator, PDFP²O) 来求解问题 (1.2). 特别地, 当 $f(x) = \frac{1}{2}\|Ax - b\|_2^2$ 时, 原始对偶不动点算法等价于 Loris 和 Verhoeven [28] 提出的广义软阈值算法. 注意到, PDFP²O 算法可以看作向前向后分裂算法与计算函数 $h \circ B$ 的邻近算子的不动点算法 [27] 的结合, 因此, 它也属于原始对偶方法的范畴. 文献 [29] 对原始对偶相关算法做了比较全面的回顾和总结.

近年来, 三个凸函数和的极小化问题同样备受关注, 具体形式是

$$\min_{x \in X} f(x) + g(x) + h(Bx), \quad (1.3)$$

其中 X 和 Y 是给定实 Hilbert 空间, $f : X \rightarrow \mathbb{R}$ 是凸可微函数且其梯度满足 Lipschitz 连续性, $g \in \Gamma_0(X)$ 和 $h \in \Gamma_0(Y)$ 可能是非光滑的, $B : X \rightarrow Y$ 是有界线性算子. 与前类似, 总是假设函数 g 和 h 是简单的. 很明显, 问题 (1.3) 包含了前面所述问题 (1.1) 和 (1.2). 虽然前面提到的迭代算法可以被用来求解所考虑的三个凸函数和的问题 (1.3), 但相应的迭代算法会引入更多的对偶变量或者未利用函数 $f(x)$ 的可微性. 为了克服这些缺点, 一些有效的迭代算法被提出用来求解优化问题 (1.3). Condat [30] 提

出了原始对偶分裂算法求解问题 (1.3). Vu^[31] 提出了用于解决单调包含零点问题的迭代算法, 它是基于文献 [32] 中提出的原始对偶分裂框架. 当 Vu 提出的迭代算法应用于求解优化问题 (1.3) 时, 可以发现 Vu 算法^[31] 与 Condat 算法^[30] 基本是一致的. 在下文中, 用 Condat-Vu 算法来引用 Condat^[30] 和 Vu^[31] 独立提出的迭代算法. Condat-Vu 算法包含了几类现有迭代算法, 例如, 当 $f = 0$ 时, 它退化成了 Chambolle 和 Pock^[17] 提出的原始对偶邻近点算法; 当 $h = 0$ 时, 它退化为向前向后分裂算法^[1]. Condat-Vu 算法的迭代参数受到算子范数 B 和 ∇f 的 Lipschitz 常数同时约束, 这在一定程度上使得参数选择变得困难. 对于 Condat-Vu 算法, Li 和 Zhang^[33] 提出了广义原始对偶分裂算法, 并证明了算法的收敛性, 同时, 他们证明了该迭代算法在遍历意义和部分原始对偶间隙意义下具有 $O(1/k)$ 的收敛率. 进而他们还引入了拟 Newton 和超松弛策略来加速所提算法. Wen 等^[34] 基于预方法思想^[35] 提出了自适应原始对偶分裂算法, 该算法的优点是无需计算算子 B 的范数, 同时所得算法比原始对偶分裂算法具有更快的收敛速度. 此外, Latafat 和 Patrinos^[36] 提出了不对称向前向后共轭 (asymmetric forward-backward-adjoint splitting, AFBA) 分裂算法, 推广了 Condat-Vu 算法. 基于原始对偶不动点算法^[26,28] 和预条件交替投影算法 (preconditioned alternating projection algorithm, PAPA)^[37] 的思想, Chen 等^[38] 提出了一种原始对偶不动点 (PDFP) 算法来求解优化问题 (1.3); 并借助不动点理论, 证明了 PDFP 算法的收敛性, 同时在适当的条件下证明了迭代算法的收敛率. 最近, Yan^[39] 提出了一种新的三算子原始对偶 (PD3O) 算法来求解优化问题 (1.3), 并指出当 $f = 0$ 时, PD3O 算法退化为 Chambolle 和 Pock 的原始对偶邻近点算法^[17,35]; 当 $g = 0$ 时, PD3O 算法退化为 PDFP²O 算法^[26]. 此外, 当 B 是恒等算子时, 它恢复了由 Davis 和 Yin^[40] 提出的三算子分裂算法.

据我们所知, Condat-Vu 算法^[30,31]、PDFP 算法^[38] 和 PD3O 算法^[39] 是三种主要用来求解三个凸函数和的优化问题 (1.3) 的算法. 然而, 现有的结果并没有充分说明这些迭代算法与其他迭代算法的联系. 本文将提出一种内外迭代算法来求解优化问题 (1.3), 这些算法主要基于向前向后分裂算法^[1] 和三算子分裂算法^[40]. 我们的工作主要分为两部分: (1) 以向前向后分裂算法为基础, 为了计算函数 $g + h \circ B$ 的邻近算子, 注意到在一般情形下, 该函数的邻近算子没有封闭形式的显示解. 为了解决这个难题, 我们采用了两种方法: 一种是对偶方法, 另一种是原始对偶方法. 在对偶方法中, 首先推导出求该函数邻近算子的极小化问题的对偶问题, 由于该对偶问题是两个凸函数的和, 其中一个是连续可微的, 另一个是 h 的 Fenchel 共轭. 然后由向前向后分裂算法求解该对偶问题, 进而通过对偶问题最优化解得到原始问题最优解. 在原始对偶方法中, 我们基于原始对偶邻近点算法求解函数 $g + h \circ B$ 的邻近算子极小化问题. 从而, 我们提出两种内外迭代形式的算法求解优化问题 (1.3), 并指出所提出的迭代算法与 PDFP 算法^[38] 和 Condat-Vu 算法^[30,31] 的联系. (2) 我们采用三算子分裂算法求解所考虑的优化问题 (1.3). 在三算子迭代算法中, 我们需要计算函数 $h \circ B$ 的邻近算子. 类似前面求解函数 $g + h \circ B$ 的邻近算子方法, 我们分别使用对偶和原始对偶方法来求解函数 $h \circ B$ 的邻近算子. 我们同样建立两种内外迭代形式的算法求解优化问题 (1.3), 并说明 PD3O 算法^[39] 与我们提出的迭代算法之间的联系. 我们在向前向后分裂算法^[1] 和三算子分裂算法^[40] 的框架下, 分别证明了所提出的 4 种具体内外迭代算法的收敛性. 为了验证所提出迭代算法的有效性, 我们分别应用于求解融合 Lasso 问题 (1.4)、约束全变分正则化问题 (1.5) 和低秩全变分图像超分辨率重建问题 (1.7). 根据数值计算结果, 我们对如何选择所提出的迭代算法的参数进行详细的讨论. 下面简要回顾这三类问题.

(1) 融合 Lasso 问题 (fused Lasso). 融合 Lasso 问题是由 Tibshirani 等^[41] 首次提出的, 它是 Lasso 问题的扩展. 该模型不仅对回归系数进行惩罚, 而且对相邻变量的回归系数之差的绝对值进行惩罚. 具体形式如下:

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} \|Ax - b\|_2^2 + \mu_1 \|x\|_1 + \mu_2 \|Dx\|_1, \quad (1.4)$$

其中 $\mu_1 > 0$ 和 $\mu_2 > 0$ 是正则参数, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^n$ 以及差分矩阵 $D_{(n-1) \times n}$ 满足

$$D = \begin{pmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{pmatrix}.$$

融合 Lasso 问题具有广泛的应用性, 可参见文献 [42–45]. 当 $A = I$ 时, 融合 Lasso 问题变成 $\mu_1 \|x\|_1 + \mu_2 \|Dx\|_1$ 的邻近算子问题, 它最先由 Friedman 等 [46] 研究. Liu 等 [43] 简化了 Friedman 等 [46] 的方法. 最近, Shi 等 [47] 考虑了更加一般的 $g(x) + \mu_2 \|Dx\|_1$ 的邻近算子, 其中 $g \in \Gamma_0(X)$, 所得结果推广了 Friedman 等 [46] 和 Liu 等 [43] 的结果. 注意到我们考虑的问题 (1.3) 比文献 [43, 46, 47] 中所研究的问题更具有一般性.

(2) 约束全变分正则化问题. TV 正则化 [8] 已经被普遍应用于图像复原和图像重建等问题中 (如文献 [48, 49] 等). 为了求得恢复图像, 考虑求解下面的约束全变分正则化问题:

$$\min_{x \in C} \frac{1}{2} \|Ax - b\|_2^2 + \mu \|x\|_{\text{TV}}, \quad (1.5)$$

等价于

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} \|Ax - b\|_2^2 + \mu \|x\|_{\text{TV}} + \delta_C(x), \quad (1.6)$$

其中 $A \in \mathbb{R}^{m \times n}$ 是系统矩阵, $b \in \mathbb{R}^m$ 是受到噪声污染的采集数据, $\mu > 0$ 是正则参数, 这里正则参数是用来保持数据误差项与正则项之间的平衡性. C 是根据图像的先验信息而设置的非空闭凸集, 如非负集合或有界盒子集合等. δ_C 表示闭凸集 C 的指示函数, 满足如果 $x \in C$, 则 $\delta_C(x) = 0$, 否则 $\delta_C(x) = +\infty$. 注意到, 全变分 $\|x\|_{\text{TV}}$ 可以表示为凸函数和离散梯度算子的复合形式 (见定义 5.1).

(3) 低秩全变分图像超分辨率重建问题. 低秩正则约束同样在图像恢复等问题中有着广泛的应用. 考虑低秩全变分 (low-rank total variation, LRTV) 图像超分辨率重建问题,

$$\min_{X \in \mathbb{R}^{m \times n}} \frac{1}{2} \|DSX - T\|_F^2 + \lambda_1 \|X\|_* + \lambda_2 \|X\|_{\text{TV}}, \quad (1.7)$$

其中 $\lambda_1 > 0$ 和 $\lambda_2 > 0$ 是正则参数, T 表示低分辨率图像, D 是下采样算子, S 是模糊算子, X 是待恢复的高分辨率图像, $\|\cdot\|_F$ 表示矩阵的 Frobenius 范数, $\|X\|_*$ 表示矩阵 X 的核范数. 低秩全变分图像超分辨率重建模型 (1.7) 首次出现在文献 [50] 中, 并在文献 [51] 中得到了进一步深入研究. 低秩全变分图像超分辨率重建模型 (1.7) 可以看作全变分图像超分辨率模型 [52] 的推广.

本文具体安排如下. 第 2 节主要回顾凸分析中的一些基本概念和若干重要引理. 第 3 节提出基于向前向后分裂算法并结合对偶和原始对偶方法来求解所考虑的优化问题 (1.3) 的迭代算法. 我们在有限维 Hilbert 空间证明所提出迭代算法的收敛性. 同时, 我们指出现有迭代算法与我们的迭代算法之间的联系. 第 4 节借助三算子分裂算法与对偶和原始对偶方法结合来求解优化问题 (1.3). 我们证明本文所提出的迭代算法的收敛性以及说明这些迭代算法与现有算法之间的联系. 第 5 节应用所提出的迭代算法分别求解融合 Lasso 问题 (1.4)、约束全变分正则化问题 (1.5) 和低秩全变分图像超分辨率问题 (1.7). 最后给出全文总结和展望.

2 预备知识

在本文中, 设 X 是一个实 Hilbert 空间, 定义在 X 上的内积和范数分别表示为 $\langle \cdot, \cdot \rangle$ 和 $\|\cdot\|$. 设 I 是 X 上的恒等算子. 设算子 $B: X \rightarrow Y$, Y 是实 Hilbert 空间, 其范数定义为 $\|B\| = \sup_{x \in X} \frac{\|Bx\|}{\|x\|}$. 算子 B 的伴随算子 B^* 满足 $\langle x, By \rangle = \langle B^*x, y \rangle$. 设 $C \subseteq X$ 是 X 的一个非空闭凸子集, C 的内点和相对内点分别表示为 $\text{int}C$ 和 $\text{ri}C$. C 上的指示函数 $\delta_C(x)$ 定义为

$$\delta_C(x) = \begin{cases} 0, & x \in C, \\ +\infty, & \text{其他}. \end{cases} \quad (2.1)$$

设函数 $f: X \rightarrow (-\infty, +\infty]$, 若 $\lim_{\|x\| \rightarrow +\infty} f(x) = +\infty$, 则称实值函数 f 是强制的. $\text{dom}f = \{x \in X : f(x) < +\infty\}$ 为 f 的定义域. 如果 $\text{dom}f \neq \emptyset$, 则称 f 是正则的. 如果 $\liminf_{x \rightarrow x_0} f(x) \geq f(x_0)$, 则实值函数 f 是下半连续的. f 的次微分是集值算子 $\partial f: X \rightarrow 2^X : x \mapsto \{u \in X \mid f(y) \geq f(x) + \langle u, y-x \rangle, \forall y \in X\}$. $\Gamma_0(X)$ 表示所有定义在 $X \rightarrow (-\infty, +\infty]$ 的下半连续 (lower semi-continuous, LSC) 正则凸函数的全体.

以下事实将是必需的.

引理 2.1 ^[53] 设 $g \in \Gamma_0(X)$, $h \in \Gamma_0(Y)$, $B: X \rightarrow Y$ 是一个有界线性算子而且 $0 \in \text{int}(B(\text{dom}g) - \text{dom}h)$, 则 $\partial(g + h \circ B) = \partial g + B^* \circ \partial h \circ B$.

我们将使用 Fenchel 共轭的概念来推导出所考虑的优化问题的对偶问题, 下面简要回顾它的定义.

定义 2.1 (Fenchel 共轭) 设 $f: X \rightarrow (-\infty, \infty]$, f 的 Fenchel 共轭表示为 f^* , 定义如下:

$$f^*(u) = \sup_x \langle x, u \rangle - f(x). \quad (2.2)$$

当 f 是一个正则的下半连续凸函数时, $f^{**} = f$. 进一步, 有 $u \in \partial f(x) \Leftrightarrow x \in \partial f^*(u)$.

1962 年, Moreau ^[54] 引入了邻近算子的概念, 它在解决凸优化问题的邻近点算法中起着非常重要的作用. 关于邻近点算法的综述可以参见文献 [55].

定义 2.2 (邻近算子) 设 $f: X \rightarrow (-\infty, \infty]$ 是一个正则的下半连续凸函数. 设 $\lambda > 0$, λf 的邻近算子 $\text{prox}_{\lambda f}: X \rightarrow X$ 的定义是

$$\text{prox}_{\lambda f}(v) = \arg \min_x \left\{ \frac{1}{2\lambda} \|x - v\|^2 + f(x) \right\}. \quad (2.3)$$

邻近算子有许多重要的性质. 例如, $\text{prox}_{\lambda f}$ 是固定非扩张的, 即

$$\|\text{prox}_{\lambda f}(x) - \text{prox}_{\lambda f}(y)\|^2 + \|(x - \text{prox}_{\lambda f}(x)) - (y - \text{prox}_{\lambda f}(y))\|^2 \leq \|x - y\|^2, \quad \forall x, y \in X.$$

上述不等式等价于

$$\|\text{prox}_{\lambda f}(x) - \text{prox}_{\lambda f}(y)\|^2 \leq \langle x - y, \text{prox}_{\lambda f}(x) - \text{prox}_{\lambda f}(y) \rangle, \quad \forall x, y \in X.$$

借助 Cauchy-Schwarz 不等式, 可以得出邻近算子也是非扩张的, 即

$$\|\text{prox}_{\lambda f}(x) - \text{prox}_{\lambda f}(y)\| \leq \|x - y\|, \quad \forall x, y \in X.$$

邻近算子是正交投影算子的推广. 事实上, 令 $f(x) = \delta_C(x)$, 则邻近算子 $\text{prox}_{\lambda \delta_C}(v) = P_C(v)$. ℓ_1 -norm 的邻近算子是著名的软门限算子, 已经被广泛应用于稀疏优化问题. 关于其他函数的邻近算子形式, 可以参见文献 [56].

定义 2.3 (Moreau 包络) 设 $f \in \Gamma_0(X)$ 和 $\lambda > 0$, Moreau 包络 (Moreau-Yosida 正则化) 定义是

$$\tilde{f}_\lambda(x) = \inf_y \left\{ f(y) + \frac{1}{2\lambda} \|x - y\|^2 \right\}. \quad (2.4)$$

下面一个引理表明 Moreau-Yosida 正则化在 X 上是可微的.

引理 2.2 ^[56] 设 $\lambda > 0$ 和 $f \in \Gamma_0(X)$, 则函数 $\tilde{f}_\lambda(X) : X \rightarrow \mathbb{R}$ 是可微的并且它的梯度是

$$\nabla \tilde{f}_\lambda(x) = \frac{1}{\lambda}(x - \text{prox}_{\lambda f}(x)), \quad (2.5)$$

且它是 $\frac{1}{\lambda}$ -Lipschitz 连续的.

下面的 Moreau 恒等式表明, 可以从它的 Fenchel 共轭 f^* 来计算正则的下半连续凸函数 f 的邻近算子, 反之也成立.

引理 2.3 ^[56] 对任意的 $\lambda > 0$, 设向量 $u \in X$, Moreau 恒等式满足

$$\text{prox}_{\lambda f}(u) + \lambda \text{prox}_{\frac{1}{\lambda} f^*} \left(\frac{1}{\lambda} u \right) = u. \quad (2.6)$$

通过 Fenchel 共轭和邻近算子的定义, 我们可以证明下面的引理.

引理 2.4 设 $\lambda > 0$, $f \in \Gamma_0(X)$, 对任意的 $u \in X$, 有

$$\text{prox}_{(\lambda f)^*}(u) = \lambda \text{prox}_{\frac{1}{\lambda} f^*} \left(\frac{1}{\lambda} u \right). \quad (2.7)$$

证明 由 Fenchel 共轭的定义, 对任意的 $u \in X$, 有

$$\begin{aligned} (\lambda f)^*(u) &= \sup_x \langle x, u \rangle - (\lambda f)(x) \\ &= \lambda \sup_x \left\langle x, \frac{1}{\lambda} u \right\rangle - f(x) \\ &= \lambda f^* \left(\frac{1}{\lambda} u \right). \end{aligned} \quad (2.8)$$

利用上述等式关系, 并结合邻近算子的定义, 可得

$$\begin{aligned} \text{prox}_{(\lambda f)^*}(u) &= \arg \min_x \left\{ \frac{1}{2} \|x - u\|^2 + (\lambda f)^*(x) \right\} \\ &= \arg \min_x \left\{ \frac{1}{2} \|x - u\|^2 + \lambda f^* \left(\frac{1}{\lambda} x \right) \right\}. \end{aligned} \quad (2.9)$$

令 $x' = \frac{1}{\lambda} x$, 则有

$$\begin{aligned} \min_x \frac{1}{2} \|x - u\|^2 + \lambda f^* \left(\frac{1}{\lambda} x \right) &= \min_{x'} \frac{1}{2} \|\lambda x' - u\|^2 + \lambda f^*(x') \\ &= \min_{x'} \lambda^2 \left(\frac{1}{2} \left\| x' - \frac{1}{\lambda} u \right\|^2 + \frac{1}{\lambda} f^*(x') \right). \end{aligned} \quad (2.10)$$

因此, $\text{prox}_{(\lambda f)^*}(u) = \lambda \text{prox}_{\frac{1}{\lambda} f^*} \left(\frac{1}{\lambda} u \right)$. □

我们将充分利用以下引理来推导我们的迭代算法. 这个引理在文献 [57] 中被证明. 同时也能在文献 [56] 中看到.

引理 2.5^[57] 设 $g \in \Gamma_0(X)$, $h \in \Gamma_0(Y)$, $u \in X$, $r \in Y$ 并且 $B: X \rightarrow Y$ 是满足 $r \in \text{int}(B(\text{dom } g) - \text{dom } h)$ 的一个有界线性算子. 考虑最小化问题

$$\min_{x \in X} \frac{1}{2} \|x - u\|^2 + g(x) + h(Bx - r), \quad (2.11)$$

则

(i) (2.11) 的对偶问题是

$$\max_{y \in Y} -\frac{1}{2} \|B^*y - u\|^2 + \tilde{g}_1(u - B^*y) - h^*(y) - \langle y, r \rangle + \frac{1}{2} \|u\|^2. \quad (2.12)$$

(ii) 令 y^* 是对偶问题 (2.12) 的最优解并且 $x^* = \text{prox}_g(u - B^*y^*)$. 于是, x^* 是原始问题 (2.11) 的最优解.

(iii) 原始问题 (2.11) 的最优值等于对偶问题 (2.12) 的最优值.

以下引理在文献 [33] 中被证明.

引理 2.6^[33] 设 $f_1 \in \Gamma_0(X)$ 和 $f_2 \in \Gamma_0(Y)$, 假设有界线性算子 $B: X \rightarrow Y$ 满足 $0 \in \text{int}(B(\text{dom } f_1) - \text{dom } f_2)$. 考虑以下一般凸优化问题:

$$\min_x f_1(x) + f_2(Bx). \quad (2.13)$$

令 x 是 (2.13) 的解, 则对任意的 $\sigma > 0$, $\tau > 0$, 存在一个向量 $y \in Y$, 有

$$\begin{aligned} x &= \text{prox}_{\tau f_1}(x - \tau B^*y), \\ y &= \text{prox}_{\sigma f_2^*}(y + \sigma Bx). \end{aligned} \quad (2.14)$$

反之, 如果存在 $\sigma > 0$, $\tau > 0$, $x \in X$ 和 $y \in Y$ 满足 (2.14), 则 x 是 (2.13) 的解.

3 向前向后算子分裂算法求解 (1.3)

本节将提出有效的迭代算法用来解决优化问题 (1.3). 我们做以下两个假设: (1) 优化问题 (1.3) 的最小解总是存在的; (2) $0 \in \text{int}(B(\text{dom } g) - \text{dom } h)$. 正如在引言部分提到的, 在优化问题 (1.3) 中, 因为 $f(x)$ 是光滑的, 所以可以使用具有误差的向前向后分裂算法^[1]. 具体迭代算法格式如下: 对任意 $x^0 \in X$, 定义

$$x^{k+1} = \text{prox}_{\gamma(g+h \circ B)}(x^k - \gamma \nabla f(x^k)) + e_k, \quad (3.1)$$

其中 $\gamma \in (0, 2/L)$, L 是 ∇f 的 Lipschitz 常数, e_k 为迭代更新序列 x^{k+1} 与邻近算子 $\text{prox}_{\gamma(g+h \circ B)}(x^k - \gamma \nabla f(x^k))$ 之间的某种误差. 关于具有误差的向前向后分裂算法的收敛性证明可以参见文献 [58].

定理 3.1^[58] 设 $\gamma \in (0, 2/L)$, 其中 L 是 ∇f 的 Lipschitz 常数. 对任意 $x^0 \in X$, 迭代序列 $\{x^k\}$ 由 (3.1) 定义. 假设 $\sum_{k=0}^{\infty} \|e_k\| < +\infty$, 则迭代序列 $\{x^k\}$ 弱收敛到优化问题 (1.3) 的一个解.

因为函数 $\gamma(g + h \circ B)$ 的邻近算子没有封闭的显式解, 因此不能获得迭代更新序列 $\{x^{k+1}\}$ 的具有误差 $e_k = 0$ 的精确值. 然而, 如果可以得到邻近算子的一个充分精确的近似值且使得误差向量 $\{e_k\}$ 满足定理 3.1 的要求, 则所得到的迭代序列将收敛到优化问题 (1.3) 的解. 因此, 为了获得迭代序列 $\{x^{k+1}\}$, 关键的问题是如何有效地计算邻近算子 $\gamma(g + h \circ B)$, 即 $\text{prox}_{\gamma(g+h \circ B)}$. 在本文中, 考虑两种有效的算法求解函数 $\gamma(g + h \circ B)$ 在点 $x^k - \gamma \nabla f(x^k)$ 的邻近算子: 一种是对偶方法, 另外一种是对原始对偶方法. 这两种算法都可以得到邻近算子 $\text{prox}_{\gamma(g+h \circ B)}(x^k - \gamma \nabla f(x^k))$ 的一个近似解同时使得误差满足定理 3.1 的要求.

3.1 对偶方法

首先, 考虑使用对偶方法. 回顾邻近算子 $\text{prox}_{\gamma(g+h \circ B)}(x^k - \gamma \nabla f(x^k))$ 的定义, 它等价于求解如下最优化问题:

$$\min_{x \in X} \left\{ \frac{1}{2} \|x - (x^k - \gamma \nabla f(x^k))\|^2 + \gamma g(x) + \gamma h(Bx) \right\}. \quad (3.2)$$

易见, 问题 (3.2) 恰好是问题 (2.11) 的一个特殊的情形. 事实上, 在引理 2.5 中, 令 $r = 0$, $u = x^k - \gamma \nabla f(x^k)$. 定义 $g := \gamma g$, $h := \gamma h$, 则可以得到最优化问题 (3.2) 的对偶形式为

$$\begin{aligned} \max_{y \in Y} & -\frac{1}{2} \|B^*y - u\|^2 + \gamma \tilde{g}_\gamma(u - B^*y) - (\gamma h)^*(y) + \frac{1}{2} \|u\|^2 \\ & = \max_{y \in Y} -\frac{1}{2} \|B^*y - u\|^2 + \gamma \tilde{g}_\gamma(u - B^*y) - \gamma h^*\left(\frac{1}{\gamma}y\right) + \frac{1}{2} \|u\|^2. \end{aligned} \quad (3.3)$$

由 $(\gamma h)^*(y) = \gamma h^*\left(\frac{1}{\gamma}y\right)$ 可知, (3.3) 中的等式成立. 设 y 是对偶问题 (3.3) 的一个解. 根据引理 2.5(ii) 可知, $x = \text{prox}_{\gamma g}(u - B^*y)$ 是优化问题 (3.2) 的一个解. 在 (3.3) 中, 令 $y' = \frac{1}{\gamma}y$, 于是, 对偶问题 (3.3) 可简化为

$$\max_{y' \in Y} \gamma^2 \left(-\frac{1}{2} \left\| B^*y' - \frac{1}{\gamma}u \right\|^2 + \frac{1}{\gamma} \tilde{g}_\gamma(u - \gamma B^*y') - \frac{1}{\gamma} h^*(y') + \frac{1}{2\gamma^2} \|u\|^2 \right). \quad (3.4)$$

相对应的优化问题 (3.2) 的解 x 满足 $x = \text{prox}_{\gamma g}(u - \gamma B^*y')$. 设

$$F(y') = \frac{1}{2} \left\| B^*y' - \frac{1}{\gamma}u \right\|^2 - \frac{1}{\gamma} \tilde{g}_\gamma(u - \gamma B^*y'),$$

则对偶问题 (3.4) 等价于下面最优化问题:

$$\min_{y' \in Y} F(y') + \frac{1}{\gamma} h^*(y'). \quad (3.5)$$

由引理 2.2, 有 $\nabla F(y') = -\frac{1}{\gamma} B \text{prox}_{\gamma g}(u - \gamma B^*y')$, $\nabla F(y')$ 是 Lipschitz 连续的, 其中 Lipschitz 常数为 $\lambda_{\max}(BB^*)$, 记 $\lambda_{\max}(BB^*)$ 是算子 BB^* 的谱半径. 事实上, 因为邻近算子是非扩张的, 则对于任意的 $y_1, y_2 \in Y$, 有

$$\begin{aligned} \|\nabla F(y_1) - \nabla F(y_2)\| &= \left\| \frac{1}{\gamma} B \text{prox}_{\gamma g}(u - \gamma B^*y_1) - \frac{1}{\gamma} B \text{prox}_{\gamma g}(u - \gamma B^*y_2) \right\| \\ &\leq \frac{1}{\gamma} \|B\| \|\text{prox}_{\gamma g}(u - \gamma B^*y_1) - \text{prox}_{\gamma g}(u - \gamma B^*y_2)\| \\ &\leq \frac{1}{\gamma} \|B\| \|(\gamma B^*y_1 - \gamma B^*y_2)\| \\ &\leq \lambda_{\max}(BB^*) \|y_1 - y_2\|. \end{aligned} \quad (3.6)$$

基于向前向后分裂算法, 运用下面的迭代算法来求解 (3.5). 对任意 $y^0 \in Y$, 选择 $0 < \lambda < \frac{2}{\lambda_{\max}(BB^*)}$, 可得

$$y^{j_k+1} = \text{prox}_{\frac{\lambda}{\gamma} h^*} \left(y^{j_k} + \frac{\lambda}{\gamma} B \text{prox}_{\gamma g}(u - \gamma B^*y^{j_k}) \right), \quad j_k = 0, 1, 2, \dots, \quad (3.7)$$

从而提出下面的迭代算法求解优化问题 (1.3), 其中迭代算法包括一个外迭代步骤和一个内迭代步骤.

算法 3.1 一种对偶向前向后分裂算法求解优化问题 (1.3)

初始化: 任取 $x^0 \in X$, $y^0 \in Y$. 选择 $\gamma \in (0, 2/L)$, $\lambda \in (0, 2/\lambda_{\max}(BB^*))$.

(1) (外迭代步骤) 当 $k = 0, 1, 2, \dots$ 时,

$$u^k = x^k - \gamma \nabla f(x^k).$$

(2) (内迭代步骤) 当 $j_k = 0, 1, 2, \dots$ 时,

$$y^{j_k+1} = \text{prox}_{\frac{\lambda}{\gamma}h^*}(y^{j_k} + \frac{\lambda}{\gamma}B\text{prox}_{\gamma g}(u^k - \gamma B^*y^{j_k})).$$

当满足停止条件时, 终止内迭代步骤. 输出: y^{j_k} .

(3) $x^{k+1} = \text{prox}_{\gamma g}(u^k - \gamma B^*y^{j_k})$.

(4) 当满足停止条件时, 终止外迭代步骤.

下面在有限维的 Hilbert 空间中证明算法 3.1 的收敛性.

定理 3.2 设 $\gamma \in (0, 2/L)$, $\lambda \in (0, 2/\lambda_{\max}(BB^*))$. 对任意 $x^0 \in X$, $y^0 \in Y$, 迭代序列 $\{x^k\}$ 和 $\{y^k\}$ 由算法 3.1 定义, 则迭代序列 $\{x^k\}$ 收敛到优化问题 (1.3) 的一个解.

证明 设 y 是优化问题 (3.5) 的一个解. 根据引理 2.5 可知, $x = \text{prox}_{\gamma g}(u^k - \gamma B^*y)$ 是 (3.2) 的一个解. 根据向前向后分裂算法的收敛性, 当 $k \rightarrow \infty$ 时, 有 $\|y^k - y\| \rightarrow 0$. 因此, 对于给定的常数 $1/(\gamma\|B\|k^2) > 0$, 存在一个整数 j_k , 当 $J_k \geq j_k$ 时, $\|y^{J_k} - y\| \leq 1/(\gamma\|B\|k^2)$. 注意到 $x^{k+1} = \text{prox}_{\gamma g}(u^k - \gamma B^*y^{J_k})$, 可得

$$\begin{aligned} \|e_k\| &= \|x^{k+1} - x\| \\ &= \|\text{prox}_{\gamma g}(u^k - \gamma B^*y^{J_k}) - \text{prox}_{\gamma g}(u^k - \gamma B^*y)\| \\ &\leq \|\gamma B^*y^{J_k} - \gamma B^*y\| \leq \frac{1}{k^2}, \end{aligned} \quad (3.8)$$

其中第一个不等式是根据邻近算子是非扩张的, 第二个不等式由于 y^{J_k} 和 y 的估计, 则 $\sum_{k=0}^{\infty} \|e_k\| < +\infty$. 由定理 3.1 可以得出, 迭代序列 $\{x^k\}$ 收敛到优化问题 (1.3) 的一个解. 定理完毕. $\square$

3.2 原始对偶方法

本小节运用原始对偶方法求解优化问题 (3.2). 我们将看到原始对偶方法同样可以得到优化问题 (3.2) 的一个近似精确解. 我们主要基于文献 [17] 所提出的原始对偶邻近点算法求解 (3.2), 具体迭代算法如下: 对任意 $\bar{x}^0 \in X$, $y^0 \in Y$, 取 $\tau > 0$, $\sigma > 0$ 满足 $\tau\sigma\|B\|^2 < 1$, 令 $j_k = 0, 1, 2, \dots$, 定义

$$\bar{x}^{j_k+1} = \text{prox}_{\tau(\frac{1}{2}\|\cdot - u\|^2 + \gamma g)}(\bar{x}^{j_k} - \tau B^*y^{j_k}), \quad (3.9)$$

$$y^{j_k+1} = \text{prox}_{\sigma(\gamma h)^*}(y^{j_k} + \sigma B(2\bar{x}^{j_k+1} - \bar{x}^{j_k})). \quad (3.10)$$

根据邻近算子定义, 通过简单计算, 可得

$$\bar{x}^{j_k+1} = \text{prox}_{\frac{\tau\gamma}{1+\tau}g}\left(\frac{\bar{x}^{j_k} - \tau B^*y^{j_k} + \tau u}{1 + \tau}\right), \quad (3.11)$$

$$y^{j_k+1} = \gamma \text{prox}_{\frac{\sigma}{\gamma}h^*}\left(\frac{1}{\gamma}(y^{j_k} + \sigma B(2\bar{x}^{j_k+1} - \bar{x}^{j_k}))\right). \quad (3.12)$$

因此, 可以得到基于原始对偶方法和向前向后分裂算法结合的迭代算法, 具体的迭代算法见算法 3.2.

算法 3.2 一种原始对偶向前向后分裂算法求解优化问题 (1.3)

初始化: 任取 $x^0, \bar{x}^0 \in X, y^0 \in Y$. 选择 $\gamma \in (0, 2/L)$. 令 $\sigma > 0, \tau > 0$ 满足条件 $\tau\sigma < \frac{1}{\|B\|^2}$.

(1) (外迭代步骤) 当 $k = 0, 1, 2, \dots$ 时,

$$u^k = x^k - \gamma \nabla f(x^k).$$

(2) (内迭代步骤) 当 $j_k = 0, 1, 2, \dots$ 时,

$$(2.1) \bar{x}^{j_k+1} = \text{prox}_{\frac{\tau\gamma}{1+\tau}g} \left(\frac{\bar{x}^{j_k} - \tau B^* y^{j_k} + \tau u^k}{1+\tau} \right);$$

$$(2.2) y^{j_k+1} = \gamma \text{prox}_{\frac{\sigma}{\gamma}h^*} \left(\frac{1}{\gamma} (y^{j_k} + \sigma B(2\bar{x}^{j_k+1} - \bar{x}^{j_k})) \right).$$

当满足停止条件时, 终止内迭代并且输出: $\bar{x}^{J_k}$.

(3) 更新 $x^{k+1} = \bar{x}^{J_k}$.

(4) 当满足停止条件时, 终止外迭代.

下面在有限维 Hilbert 空间中证明算法 3.2 的收敛性.

定理 3.3 设 $\gamma \in (0, 2/L)$, 选择 $\sigma > 0$ 和 $\tau > 0$ 满足条件 $\tau\sigma < \frac{1}{\|B\|^2}$. 对任意 $x^0, \bar{x}^0 \in X$ 以及 $y^0 \in Y$, 迭代序列 $\{x^k\}$ 、 $\{\bar{x}^k\}$ 和 $\{y^k\}$ 是由算法 3.2 定义, 则迭代序列 $\{x^k\}$ 收敛到优化问题 (1.3) 的一个解.

证明 在引理 2.6 中, 设 $f_1(x) = \frac{1}{2}\|x - u^k\|^2 + \gamma g(x)$, $f_2(x) = \gamma h(x)$. 设 x 是 (3.2) 的一个解. 由引理 2.6 可知, 对任意 $\tau > 0, \sigma > 0$, 存在一个向量 y 使得

$$\begin{aligned} x &= \text{prox}_{\tau f_1}(x - \tau B^* y), \\ y &= \text{prox}_{\sigma f_2}(y + \sigma B y). \end{aligned} \quad (3.13)$$

根据邻近算子的定义, 方程 (3.13) 可简化为

$$\begin{aligned} x &= \text{prox}_{\frac{\gamma\tau}{1+\tau}g} \left(\frac{x - \tau B^* y + \tau u^k}{1+\tau} \right), \\ y &= \text{prox}_{\sigma(\gamma h)^*}(y + \sigma B y). \end{aligned} \quad (3.14)$$

根据文献 [17, 定理 1] 可知, 迭代序列 $\{\bar{x}^k\}$ 和 $\{y^k\}$ 分别收敛于 x 和 y . 于是, 对任给常数 $\frac{1+\tau}{k^2}$, 存在整数 j_k , 当 $J_k \geq j_k$ 时, $\|\bar{x}^{J_k} - x\| + \tau\|B\|\|y^{J_k} - y\| \leq (1+\tau)\frac{1}{k^2}$. 从而,

$$\begin{aligned} \|e^k\| &= \|x^{k+1} - x\| = \left\| \text{prox}_{\frac{\gamma\tau}{1+\tau}g} \left(\frac{\bar{x}^{J_k} - \tau B^* y^{J_k} + \tau u^k}{1+\tau} \right) - \text{prox}_{\frac{\gamma\tau}{1+\tau}g} \left(\frac{x - \tau B^* y + \tau u^k}{1+\tau} \right) \right\| \\ &\leq \left\| \frac{\bar{x}^{J_k} - \tau B^* y^{J_k} + \tau u^k}{1+\tau} - \frac{x - \tau B^* y + \tau u^k}{1+\tau} \right\| \\ &= \frac{1}{1+\tau} \|(\bar{x}^{J_k} - x) - \tau B^*(y^{J_k} - y)\| \\ &\leq \frac{1}{1+\tau} (\|\bar{x}^{J_k} - x\| + \tau\|B\|\|y^{J_k} - y\|) \\ &\leq \frac{1}{k^2}. \end{aligned} \quad (3.15)$$

因此, $\sum_{k=0}^{\infty} \|e_k\| < +\infty$. 根据定理 3.1, 可以得到迭代序列 $\{x^k\}$ 收敛到优化问题 (1.3) 的一个解. 定理证明完毕. $\square$

3.3 与现有其他算法的联系

本小节将详细说明所提出的迭代算法与现有其他迭代算法之间的联系. 令 $j_k = k$ 并且内迭代步数为 1, 则算法 3.1 等价于 PDFP 算法^[38], 具体如下:

$$\begin{cases} v^{k+1} = \text{prox}_{\gamma g}(x^k - \gamma \nabla f(x^k) - \gamma B^* y^k), \\ y^{k+1} = \text{prox}_{\frac{\lambda}{\gamma} h^*} \left(y^k + \frac{\lambda}{\gamma} B v^{k+1} \right), \\ x^{k+1} = \text{prox}_{\gamma g}(x^k - \gamma \nabla f(x^k) - \gamma B^* y^{k+1}). \end{cases} \quad (3.16)$$

根据 Moreau 恒等式 (引理 2.3), (3.16) 中迭代序列 y^{k+1} 等价于

$$\begin{aligned} y^{k+1} &= \text{prox}_{\frac{\lambda}{\gamma} h^*} \left(\frac{\lambda}{\gamma} \left(\frac{\gamma}{\lambda} y^k + B v^{k+1} \right) \right) \\ &= \frac{\lambda}{\gamma} (I - \text{prox}_{\frac{\lambda}{\gamma} h}) \left(\frac{\gamma}{\lambda} y^k + B v^{k+1} \right). \end{aligned} \quad (3.17)$$

令 $\bar{y}^k = \frac{\gamma}{\lambda} y^k$, 则迭代算法 PDFP (3.16) 可改写为

$$\begin{cases} v^{k+1} = \text{prox}_{\gamma g}(x^k - \gamma \nabla f(x^k) - \lambda B^* \bar{y}^k), \\ \bar{y}^{k+1} = (I - \text{prox}_{\frac{\lambda}{\gamma} h})(y^k + B v^{k+1}), \\ x^{k+1} = \text{prox}_{\gamma g}(x^k - \gamma \nabla f(x^k) - \lambda B^* \bar{y}^{k+1}). \end{cases} \quad (3.18)$$

假设 $0 < \lambda < 1/\lambda_{\max}(BB^*)$ 和 $0 < \gamma < 2/L$ 的条件下, Chen 等^[38] 证明了迭代算法 (3.16) 的收敛性. 与 PDFP 算法^[38] 相比, 本文所提出的算法 3.1 允许迭代参数 λ 的取值范围更广. 注意到当 $f(x) = \frac{1}{2} \|x - u\|_2^2$, 同时迭代参数 $\gamma = 1$ 时, 迭代算法 PDFP 和 Combettes 等^[57] 所提出的对偶向前向后分裂算法是相同的.

接下来说明算法 3.2 与 Condat-Vu 算法之间的联系. 令 $\bar{x}^0 = x^0$, $j_k = k$ 并且固定内迭代步数是 1, 则算法 3.2 化简为

$$\begin{cases} x^{k+1} = \text{prox}_{\frac{\tau\gamma}{1+\tau} g} \left(\frac{x^k - \tau B^* y^k + \tau(x^k - \gamma \nabla f(x^k))}{1 + \tau} \right), \\ y^{k+1} = \gamma \text{prox}_{\frac{\sigma}{\gamma} h^*} \left(\frac{1}{\gamma} y^k + \frac{\sigma}{\gamma} B(2x^{k+1} - x^k) \right). \end{cases} \quad (3.19)$$

令 $\bar{y}^k = \frac{1}{\gamma} y^k$, 并且经过简单计算后, 迭代算法 (3.19) 可以等价表示为

$$\begin{cases} x^{k+1} = \text{prox}_{\frac{\tau\gamma}{1+\tau} g} \left(x^k - \frac{\tau\gamma}{1+\tau} B^* \bar{y}^k - \frac{\tau\gamma}{1+\tau} \nabla f(x^k) \right), \\ \bar{y}^{k+1} = \text{prox}_{\frac{\sigma}{\gamma} h^*} \left(\bar{y}^k + \frac{\sigma}{\gamma} B(2x^{k+1} - x^k) \right). \end{cases} \quad (3.20)$$

令 $\sigma' = \frac{\sigma}{\gamma}$, $\tau' = \frac{\tau\gamma}{1+\tau}$, 于是, 上述迭代算法 (3.20) 等价于 Condat^[30] 提出的原始对偶分裂算法:

$$\begin{cases} x^{k+1} = \text{prox}_{\tau' g}(x^k - \tau' B^* \bar{y}^k - \tau' \nabla f(x^k)), \\ \bar{y}^{k+1} = \text{prox}_{\sigma' h^*}(\bar{y}^k + \sigma' B(2x^{k+1} - x^k)). \end{cases} \quad (3.21)$$

在满足条件 $\frac{1}{\tau'} - \sigma' \|B\|^2 > L/2$ 下, Condat^[30] 证明了所提出的迭代算法 (3.21) 收敛到优化问题 (1.3) 的一个解. 在算法 3.2 中, 上述参数条件同样成立. 事实上, 因为 $\sigma = \gamma\sigma'$, $\tau = \frac{\tau'}{\gamma - \tau'}$, 根据 $\sigma\tau\|B\|^2 < 1$, $0 < \gamma < 2/L$ 的条件, 有

$$\begin{aligned} \gamma\sigma' \frac{\tau'}{\gamma - \tau'} \|B\|^2 &< 1 \\ \Leftrightarrow \frac{1}{\tau'} - \sigma' \|B\|^2 &> \frac{1}{\gamma} > \frac{L}{2}. \end{aligned} \quad (3.22)$$

在以上不等式中, 所有迭代参数是混合的, 使得参数的选取更加困难. 而在我们提出的算法 3.2 中, 选取参数更加方便, 同时我们参数选取的规则结合了向前向后分裂算法和原始对偶方法参数条件. 注意到 Yan^[39] 提出了 Condat-Vu 算法的一种等价形式如下:

$$\begin{cases} x^{k+1} = \text{prox}_{\gamma g}(x^k - \gamma B^* \bar{y}^k - \gamma \nabla f(x^k)), \\ \bar{y}^{k+1} = \text{prox}_{\frac{\sigma}{\gamma} h^*}(\bar{y}^k + \frac{\sigma}{\gamma} B(2x^{k+1} - x^k)). \end{cases} \quad (3.23)$$

在 $0 < \gamma < 2/L$ 和 $0 < \sigma < 1/\|B\|^2$ 条件下, Yan^[39] 指出迭代算法 (3.23) 可能不收敛. 在迭代参数 γ 和 σ 满足上述相同条件下, 我们给出了 Condat-Vu 算法的另外一种形式, 并且算法保持收敛. 事实上, 令 $\tau = 1$, 则迭代算法 (3.20) 可以简化为

$$\begin{cases} x^{k+1} = \text{prox}_{\frac{\gamma}{2} g}\left(x^k - \frac{\gamma}{2} B^* \bar{y}^k - \frac{\gamma}{2} \nabla f(x^k)\right), \\ \bar{y}^{k+1} = \text{prox}_{\frac{\sigma}{\gamma} h^*}\left(\bar{y}^k + \frac{\sigma}{\gamma} B(2x^{k+1} - x^k)\right). \end{cases} \quad (3.24)$$

迭代算法 (3.23) 和 (3.24) 的主要区别是: 更新迭代序列 x^{k+1} 中的迭代参数 γ 选取不同. 假设 γ 属于 $(0, 2/L)$, 迭代算法 (3.24) 收敛, 而迭代算法 (3.23) 可能不收敛.

4 三算子分裂算法求解 (1.3)

最近, Yan^[39] 提出了一种原始对偶三算子 (PD3O) 分裂算法求解优化问题 (1.3). 特别地, 当令 $B = I$ 时, PD3O 算法^[39] 便恢复了 Davis 和 Yin^[40] 提出的三算子分裂算法. 三算子分裂方法是人们所熟知的许多算子分裂方法的一种推广, 它包含了向前向后分裂方法^[59]、Douglas-Rachford 分裂方法^[60]、Peaceman-Rachford 分裂方法^[61] 和 Forward-Douglas-Rachford 分裂方法^[62]. 本节将应用三算子分裂算法求解优化问题 (1.3). 为了证明所提出迭代算法的收敛性, 我们引入具有误差的三算子分裂算法, 具体迭代格式如下: 任取 $z^0 \in X$, 有

$$\begin{aligned} x^k &= \text{prox}_{\gamma g}(z^k), \\ s^k &= \text{prox}_{\gamma(h \circ B)}(2x^k - z^k - \gamma \nabla f(x^k)) + e_k, \\ z^{k+1} &= z^k + s^k - x^k, \end{aligned} \quad (4.1)$$

其中 $\gamma \in (0, 2/L)$, e_k 表示误差向量. 关于上述三算子分裂算法, 我们有下面的收敛性定理, 具体可以参见文献 [40].

定理 4.1^[40] 假设 $\gamma \in (0, 2/L)$, L 是 ∇f 的 Lipschitz 常数. 对于任意的 $z^0 \in X$, 迭代序列 $\{x^k\}$ 、 $\{s^k\}$ 和 $\{z^k\}$ 由 (4.1) 定义. 假设 $\sum_{k=0}^{\infty} \|e_k\| < +\infty$, 则迭代序列 $\{x^k\}$ 和 $\{s^k\}$ 弱收敛到优化问题 (1.3) 的一个解.

在以上三算子迭代算法 (4.1) 中, 其关键在于求解函数 $\gamma h(Bx)$ 的邻近算子. 正如在上一节中所讨论的, 尽管它没有一个封闭形式的显示解, 但是我们仍然可以由两种方法得到它的近似精确解: 一种是对偶方法, 另一种是原始对偶方法. 接下来将具体讨论所提出迭代算法的详细过程, 并证明相应迭代算法的收敛性. 特别地, 我们将说明 PD3O 算法^[39] 与我们所提迭代算法之间的联系.

4.1 对偶方法

本小节讨论如何通过对偶方法来得到 (4.1) 中的更新迭代序列 s^k . 为了方便, 令 $u^k = 2x^k - z^k - \gamma \nabla f(x^k)$, 从而有

$$\text{prox}_{\gamma(h \circ B)}(u^k) = \arg \min_v \left\{ \frac{1}{2} \|v - u^k\|^2 + \gamma h(Bv) \right\}. \quad (4.2)$$

在引理 2.5 中, 设 $g := 0$ 和 $h := \gamma h$, 并且由 (3.3) 和 (3.4) 的推导过程, 我们得到优化问题 (4.2) 的对偶问题形式如下:

$$\max_{y \in Y} -\frac{1}{2} \|\gamma B^* y - u^k\|^2 - \gamma h^*(y). \quad (4.3)$$

同时问题 (4.2) 的原始最优解满足 $v^* = u^k - \gamma B^* y^*$, 其中 y^* 是对偶问题 (4.3) 的最优解. 注意到, 对偶最优化问题 (4.3) 能被重新改写为下面等价形式:

$$\min_{y \in Y} \frac{1}{2} \left\| B^* y - \frac{1}{\gamma} u^k \right\|^2 + \frac{1}{\gamma} h^*(y). \quad (4.4)$$

事实上, 最优化问题 (4.3) 和 (4.4) 有相同的最优解 y^* . 而优化问题 (4.4) 能够借助向前向后分裂算法来求解, 它的迭代形式如下所示: 对任意 $y^0 \in Y$, 选取 $0 < \lambda < 2/\lambda_{\max}(BB^*)$, 有

$$y^{j_k+1} = \text{prox}_{\frac{\lambda}{\gamma} h^*} \left(y^{j_k} - \lambda B \left(B^* y^{j_k} - \frac{1}{\gamma} u^k \right) \right), \quad j_k = 0, 1, 2, \dots \quad (4.5)$$

假设 $s^k = u^k - \gamma B^* y^{j_k}$, 其中 y^{j_k} 是迭代算法 (4.5) 的极限点. 这时, 对于 (4.1) 中的更新序列 z^{k+1} , 有

$$\begin{aligned} z^{k+1} &= z^k + s^k - x^k, \\ &= z^k + u^k - \gamma B^* y^{j_k} - x^k, \\ &= z^k + 2x^k - z^k - \gamma \nabla f(x^k) - \gamma B^* y^{j_k} - x^k, \\ &= x^k - \gamma \nabla f(x^k) - \gamma B^* y^{j_k}. \end{aligned} \quad (4.6)$$

因此, 基于三算子分裂方法 (4.1), 我们得到下面的迭代算法来求解优化问题 (1.3).

算法 4.1 一种对偶三算子分裂算法求解优化问题 (1.3)

初始化: 给定任意 $z^0 \in X$ 和 $y^0 \in Y$. 选择 $\gamma \in (0, 2/L)$ 和 $\lambda \in (0, 2/\lambda_{\max}(BB^*))$.

(1) (外迭代步骤) 当 $k = 0, 1, 2, \dots$ 时,

$$x^k = \text{prox}_{\gamma g}(z^k).$$

(2) (内迭代步骤) 当 $j_k = 0, 1, 2, \dots$ 时,

$$y^{j_k+1} = \text{prox}_{\frac{\lambda}{\gamma} h^*} \left((I - \lambda BB^*) y^{j_k} + \frac{\lambda}{\gamma} B(2x^k - z^k - \gamma \nabla f(x^k)) \right).$$

当停止准则达到时结束内迭代步. 输出: y^{j_k} .

$$(3) s^k = 2x^k - z^k - \gamma \nabla f(x^k) - \gamma B^* y^{j_k}.$$

$$(4) z^{k+1} = z^k + s^k - x^k.$$

(5) 当停止准则达到时, 结束外迭代步.

类似于定理 3.2, 我们将在下面给出有限维 Hilbert 空间中算法 4.1 的收敛性定理.

定理 4.2 假设 $\gamma \in (0, 2/L)$ 且 $\lambda \in (0, 2/\lambda_{\max}(BB^*))$. 对于任意的 $x^0 \in X$ 和 $y^0 \in Y$, 算法 4.1 产生迭代序列 $\{x^k\}$ 和 $\{s^k\}$, 则迭代序列 $\{x^k\}$ 和 $\{s^k\}$ 收敛到优化问题 (1.3) 的一个解.

证明 令 y 是 (4.3) 的一个最优解, 由引理 2.5 可知, $v = u^k - \gamma B^*y$ 是优化问题 (4.2) 的一个最优解, 且 $\text{prox}_{\gamma(h \circ B)}(u^k) = u^k - \gamma B^*y$. 根据经典的向前向后分裂算法的收敛性, 当 $k \rightarrow \infty$ 时, 迭代序列 $y^k \rightarrow y$. 对于给定的常数 $\frac{1}{\gamma\|B\|k^2} > 0$, 存在整数 $j_k > 0$, 当 $J_k \geq j_k$ 时, $\|y^{J_k} - y\| \leq \frac{1}{\gamma\|B\|k^2}$. 注意到 $s^k = u^k - \gamma B^*y^{J_k}$, 从而,

$$\begin{aligned} \|e_k\| &= \|s^k - \text{prox}_{\gamma(h \circ B)}(u^k)\| \\ &= \|u^k - \gamma B^*y^{J_k} - (u^k - \gamma B^*y)\| \\ &\leq \gamma\|B\|\|y^{J_k} - y\| \leq \frac{1}{k^2}. \end{aligned} \quad (4.7)$$

因此, $\sum_{k=0}^{\infty} \|e_k\| < +\infty$. 由定理 4.1 可得迭代序列 $\{x^k\}$ 和 $\{s^k\}$ 收敛到优化问题 (1.3) 的一个解. 定理证毕. $\square$

4.2 原始对偶方法

本小节利用原始对偶邻近点算法求解优化问题 (4.2). 具体迭代算法如下: 给定任意的 $v^0 \in X$ 和 $y^0 \in Y$, 当 $j_k = 0, 1, 2, \dots$ 时, 定义迭代序列

$$v^{j_k+1} = \text{prox}_{\tau(\frac{1}{2}\|\cdot - u^k\|^2)}(v^{j_k} - \tau B^*y^{j_k}), \quad (4.8)$$

$$y^{j_k+1} = \text{prox}_{\sigma(\gamma h)^*}(y^{j_k} + \sigma B(2v^{j_k+1} - v^{j_k})), \quad (4.9)$$

其中 $\tau > 0$ 和 $\sigma > 0$ 且满足 $\tau\sigma\|B\|^2 < 1$. 基于邻近算子的定义和引理 2.4, 迭代序列 (4.8) 和 (4.9) 能分别简化为

$$v^{j_k+1} = \frac{v^{j_k} - \tau B^*y^{j_k} + \tau u^k}{1 + \tau}, \quad (4.10)$$

$$y^{j_k+1} = \gamma \text{prox}_{\frac{\sigma}{\gamma} h^*} \left(\frac{1}{\gamma} y^{j_k} + \frac{\sigma}{\gamma} B(2v^{j_k+1} - v^{j_k}) \right). \quad (4.11)$$

因此, 我们可以得到如下的迭代算法来求解优化问题 (1.3).

算法 4.2 一种原始对偶三算子分裂算法求解优化问题 (1.3)

初始化: 给定任意的 $z^0, v^0 \in X$ 和 $y^0 \in Y$. 选择 $\gamma \in (0, 2/L)$. 令 $\sigma > 0$ 和 $\tau > 0$ 满足条件 $\tau\sigma < \frac{1}{\|B\|^2}$.

(1) (外迭代步骤) 当 $k = 0, 1, 2, \dots$ 时,

$$x^k = \text{prox}_{\gamma g}(z^k);$$

$$u^k = 2x^k - z^k - \gamma \nabla f(x^k).$$

(2) (内迭代步骤) 当 $j_k = 0, 1, 2, \dots$ 时,

$$(i) \ v^{j_k+1} = \frac{v^{j_k} - \tau B^*y^{j_k} + \tau u^k}{1 + \tau};$$

$$(ii) \ y^{j_k+1} = \gamma \text{prox}_{\frac{\sigma}{\gamma} h^*} \left(\frac{1}{\gamma} y^{j_k} + \frac{\sigma}{\gamma} B(2v^{j_k+1} - v^{j_k}) \right).$$

当停止准则达到时结束内迭代步. 输出: v^{J_k} .

(3) $z^{k+1} = z^k + v^{J_k} - x^k$.

(4) 当停止准则达到时, 结束外迭代步.

我们同样能证明在有限维 Hilbert 空间中算法 4.2 的收敛性. 证明方法类似于定理 3.3 的证明.

定理 4.3 假设 $\gamma \in (0, 2/L)$. 令 $\sigma > 0$ 和 $\tau > 0$ 满足条件 $\tau\sigma < \frac{1}{\|B\|^2}$. 对于任意的 $z^0, v^0 \in X$ 和 $y^0 \in Y$, 算法 4.2 产生迭代序列 $\{x^k\}$ 、 $\{z^k\}$ 、 $\{v^k\}$ 和 $\{y^k\}$, 则迭代序列 $\{x^k\}$ 和 $\{v^k\}$ 收敛到优化问题 (1.3) 的一个解.

证明 在引理 2.6 中, 定义 $f_1 := \frac{1}{2}\|\cdot - u^k\|^2$ 和 $f_2 := \gamma h$. 令 v 是 (4.2) 的最优解, 则由引理 2.6 可知, 对任意的 $\sigma > 0$ 和 $\tau > 0$, 存在一个向量 y 使得

$$\begin{cases} v = \frac{v - \tau B^*y + \tau u^k}{1 + \tau}, \\ y = \text{prox}_{\sigma(\gamma h)^*}(y + \sigma Bx). \end{cases} \quad (4.12)$$

根据原始对偶邻近点算法^[17]的收敛性, 可知 $\{v^k\}$ 和 $\{y^k\}$ 分别收敛到 v 和 y , 则对于给定的常数 $\frac{1+\tau}{k^2} > 0$, 存在整数 j_k , 当 $J_k \geq j_k$ 时, $\|v^{J_k} - v\| + \tau\|B\|\|y^{J_k} - y\| \leq \frac{1+\tau}{k^2}$. 因此,

$$\begin{aligned} \|e_k\| &= \|s^k - \text{prox}_{\gamma(h \circ B)}(u^k)\| \\ &= \left\| \frac{v^{J_k} - \tau B^*y^{J_k} + \tau u^k}{1 + \tau} - \frac{v - \tau B^*y + \tau u^k}{1 + \tau} \right\| \\ &\leq \frac{1}{1 + \tau} (\|v^{J_k} - v\| + \tau\|B\|\|y^{J_k} - y\|) \leq \frac{1}{k^2}. \end{aligned} \quad (4.13)$$

由上可知 $\sum_{k=0}^{\infty} \|e_k\| < +\infty$. 同时, 通过定理 4.1, 可得迭代序列 $\{x^k\}$ 和 $\{v^k\}$ 收敛到优化问题 (1.3) 的一个解. 定理得证. $\square$

4.3 与现有其他算法的联系

本小节先说明所提迭代算法 4.1 能够恢复 Yan^[39] 所提的 PD3O 算法. 事实上, 在算法 4.1 中, 令 $j_k = k$ 并且设内迭代数为 1, 则可得下面的 PD3O 迭代算法格式:

$$\begin{cases} x^k = \text{prox}_{\gamma g}(z^k), \\ y^{k+1} = \text{prox}_{\frac{\lambda}{\gamma} h^*} \left((I - \lambda B B^*)y^k + \frac{\lambda}{\gamma} B(2x^k - z^k - \gamma \nabla f(x^k)) \right), \\ z^{k+1} = x^k - \gamma \nabla f(x^k) - \gamma B^*y^{k+1}. \end{cases} \quad (4.14)$$

在假设条件 $0 < \gamma < 2/L$ 和 $0 < \lambda < 1/\lambda_{\max}(BB^*)$ 下, Yan^[39] 证明了 PD3O (4.14) 的收敛性. 算法 4.1 提供了比 PD3O 算法 (4.14) 更大的 λ 取值范围. 同时, Yan^[39] 也表明了求解优化问题 (1.3) 时, 当 $B = I$, PD3O (4.14) 算法等价于三算子分裂算法^[40]. 事实上, 在 PD3O (4.14) 中令 $\lambda = 1$ 和 $B = I$, 它便退化为

$$\begin{cases} x^k = \text{prox}_{\gamma g}(z^k), \\ y^{k+1} = \text{prox}_{\frac{1}{\gamma} h^*} \left(\frac{1}{\gamma} (2x^k - z^k - \gamma \nabla f(x^k)) \right), \\ z^{k+1} = x^k - \gamma \nabla f(x^k) - \gamma y^{k+1}. \end{cases} \quad (4.15)$$

根据引理 2.3 中的 Moreau 恒等式, 有

$$y^{k+1} = \text{prox}_{\frac{1}{\gamma} h^*} \left(\frac{1}{\gamma} (2x^k - z^k - \gamma \nabla f(x^k)) \right)$$

$$= \frac{1}{\gamma}(2x^k - z^k - \gamma \nabla f(x^k) - \text{prox}_{\gamma h}(2x^k - z^k - \gamma \nabla f(x^k))). \quad (4.16)$$

设 $s^k = \text{prox}_{\gamma h}(2x^k - z^k - \gamma \nabla f(x^k))$, 于是, (4.15) 等价于

$$\begin{cases} x^k = \text{prox}_{\gamma g}(z^k), \\ s^k = \text{prox}_{\gamma h}(2x^k - z^k - \gamma \nabla f(x^k)), \\ z^{k+1} = z^k + s^k - x^k. \end{cases} \quad (4.17)$$

迭代算法 (4.17) 即是文献 [40] 中所提出的精确三算子分裂算法.

接下来得到所提算法 4.2 的一种特殊情形. 假设 $j_k = k$ 且内迭代数等于 1, 则算法 4.2 的迭代格式退化为

$$\begin{cases} x^k = \text{prox}_{\gamma g}(z^k), \\ u^k = 2x^k - z^k - \gamma \nabla f(x^k), \\ v^{k+1} = \frac{v^k - \tau B^* y^k + \tau u^k}{1 + \tau}, \\ y^{k+1} = \gamma \text{prox}_{\frac{\sigma}{\gamma} h^*} \left(\frac{1}{\gamma} y^k + \frac{\sigma}{\gamma} B(2v^{k+1} - v^k) \right), \\ z^{k+1} = z^k + v^{k+1} - x^k. \end{cases} \quad (4.18)$$

所得迭代格式 (4.18) 不同于 PDFP 算法 [38]、Condat-Vu 算法 [30, 31] 和 PD3O 算法 [39]. 据我们所知, 现有的迭代算法中没有与迭代算法 (4.18) 等价的算法. 除了之前所提的三种迭代算法, 迭代格式 (4.18) 可以被看作是解决优化问题 (1.3) 的第四种迭代算法.

5 数值实验

本节通过数值实验验证所提算法的有效性, 并与现有其他算法进行比较. 我们应用所提算法求解三类问题, 分别是融合 Lasso 问题 (1.4)、具有约束的全变分正则化问题 (1.5) 和低秩全变分图像超分辨率重建问题 (1.7). 所有的实验均在联想 E4430 笔记本完成, 处理器为 Intel(R) Core(TM) i7-4712MQ CPU 2.3GHz 和 4GB 内存, 同时仿真软件为 MATLAB (2013a).

(1) 迭代算法参数选择. 一般来说, 选择越大的迭代参数 γ 将使得迭代算法的收敛越快. 因此, 在所有的迭代算法中, 选取 $\gamma = 1.9/L$, 其中 L 是 ∇f 的 Lipschitz 常数. 根据文献 [39] 等的数值实验结果可知, 迭代参数 λ 对算法的收敛速度影响比较小. 对于迭代参数 λ 的取值范围, 本文提出的迭代算法中的比 PDFP 算法 [38] 和 PD3O 算法 [39] 中的更大. 因此, 在算法 3.1 和 4.1 中, 我们考虑两种可选择的参数 λ . 此外, 对于算法 3.2 和 4.2, 我们也给出两种不同的选择 σ 和 τ 组合. 详细的迭代参数选择见表 1.

表 1 所提迭代算法中的参数选取方法

方法	参数类型 I	参数类型 II
算法 3.1 和 4.1	$\lambda_1 = 1.9/\lambda_{\max}(BB^T)$	$\lambda_2 = 1/\lambda_{\max}(BB^T)$
算法 3.2 和 4.2	$\sigma_1 = 1/\ B\ ^2, \tau_1 = 1$	$\sigma_2 = \tau_2 = 1/\ B\ $

(2) 评价标准. 为了定量评估迭代算法的性能, 我们使用信噪比 (signal-to-noise ratio, SNR) 和归一化均方距离 (normalization mean square distance, NMSD) 来衡量恢复所得信号的质量, 其中

$$\text{SNR} = 20 \log \frac{\|x - \bar{x}\|_2}{\|x - x_r\|_2}, \quad \text{NMSD} = \frac{\|x - x_r\|_2}{\|x - \bar{x}\|_2},$$

这里 x 表示原始信号, x_r 表示恢复所得信号, $\bar{x}$ 是原始信号的均值.

(3) 停止准则. 假设相对误差小于给定的数值作为迭代停止准则, 也就是

$$\frac{\|x^{k+1} - x^k\|_2}{\|x^k\|_2} \leq \epsilon,$$

其中 ϵ 是给定的一个比较小的数. 如果 x^k 表示一幅重建图像, 则上述 2-范数分别用 F -范数替代.

5.1 融合 Lasso 问题

为求解融合 Lasso 问题 (1.4), 在优化问题 (1.3) 中, 定义 $f(x) = \frac{1}{2}\|Ax - b\|_2^2$, $g(x) = \mu_1\|x\|_1$ 和 $h(Bx) = \mu_2\|Dx\|_1$, 则我们能够运用所提出的迭代算法求解融合 Lasso 问题 (1.4). 我们使用文献 [44] 的方法生成人工合成数据集, 其中原始信号 $x = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$ 满足条件

$$x_i = \begin{cases} 2, & i = 1, 2, \dots, 20, 121, 122, \dots, 125, \\ 3, & i = 41, \\ 1, & i = 71, 72, \dots, 85, \\ 0, & \text{其他.} \end{cases} \quad (5.1)$$

系统矩阵 $A \in \mathbb{R}^{m \times n}$ 是随机生成的, 它的元素服从标准的 Gauss 分布, 向量 b 满足方程 $b = Ax + e$, 其中 e 是添加的均值为 0、方差为 0.1 的 Gauss 随机噪声. 在实验中, 设 $m = 100$ 和 $n = 200$. 通过计算可知 ∇f 的 Lipschitz 常数为 $\|A\|^2$, 矩阵 DD^T 的特征值是 $2 - 2\cos(i\pi/n)$ [38], $i = 1, 2, \dots, n-1$. 这里, 取 $\lambda_{\max}(DD^T) = 4$, 正则参数分别取 $\mu_1 = 0.2$ 和 $\mu_2 = 0.8$.

在第一个数值实验中, 我们在所提出的算法 3.1、3.2、4.1 和 4.2 中分别取内迭代步数为 1. 这时, 算法 3.1 和 4.1 分别退化为相应的 FDFP 算法 [38] 和 PD3O 算法 [39], 以及算法 3.2 退化为 Condat-Vu 算法 [30, 31]. 我们按照表 1 中给定的迭代参数执行这些迭代算法. 所得数值结果见表 2. 在表 2 中, 符

表 2 求解融合 Lasso 问题 (1.4), 算法 3.1、3.2、4.1 和 4.2 所得数值结果

参数类型	方法	$\epsilon = 10^{-4}$			$\epsilon = 10^{-8}$		
		NMSD	SNR(dB)	Iter	NMSD	SNR(dB)	Iter
I	算法 3.1	0.0063	44.2488	—	0.0063	44.2488	—
	算法 3.2	0.0066	43.6446	757	0.0060	44.5044	986
	算法 4.1	0.0067	43.6584	—	0.0067	43.6584	—
	算法 4.2	0.0066	43.6411	758	0.0060	44.5044	987
II	算法 3.1	0.0060	44.4103	387	0.0060	44.5044	626
	算法 3.2	0.0076	42.3389	1119	0.0060	44.5044	1471
	算法 4.1	0.0060	44.4048	387	0.0060	44.5044	627
	算法 4.2	0.0076	42.3340	1120	0.0060	44.5044	1471

号“-”意味着相应迭代算法超过最大迭代步数 5,000. 我们能在表 2 中看出算法 3.2 和 4.2 的数值结果几乎完全相等. 当选取迭代参数类型 I 时, 算法 3.1 和 4.1 没有在给定最大迭代步数内收敛. 这是由于迭代参数类型 I 超过了算法 3.1 和 4.1 的理论 upper 要求. 当选择参数类型 II 时, 算法 3.1 和 4.1 收敛. 然而, 算法 3.2 和 4.2 相比算法 3.1 和 4.1 需要更多的迭代步数. 对于算法 3.2 和 4.2, 选择参数类型 I 的迭代步数少于选择参数类型 II 的迭代步数.

进一步, 我们在图 1 和 2 中分别给出目标函数值和 SNR 值随着迭代步数的变化曲线. 为了直观显示恢复所得信号, 图 3 展示的是恢复信号和真实的稀疏信号的比较, 其中选择的迭代参数为类型 I .

图 1 求解融合 Lasso 问题 (1.4)、目标函数值和迭代数的关系图. 选择参数类型 (a) I ; (b) II

图 2 求解融合 Lasso 问题 (1.4)、信噪比 (SNR) 值和迭代数的关系图. 选择参数类型 (a) I ; (b) II

图 3 求解融合 Lasso 问题 (1.4), 恢复所得信号与原始真实信号的比较

在第二个数值实验中, 我们说明所提迭代算法是如何受内迭代步数影响的. 我们分别选取了内迭代步数为 2、10 和 20, 选取参数类型 I 和 II , 所得数值结果分别见表 3 和 4. 当选择参数类型 I 时, 虽然在第一个实验中, 算法 3.1 和 4.1 没有收敛, 但是在表 3 中, 我们看到当内迭代步数超过 1 时, 算法 3.1 和 4.1 收敛, 同时算法 3.2 和 4.2 所需外迭代步数少于表 2 中的结果. 从表 3 和 4 中可以看出, 通过增加内迭代步数, 算法 3.2 和 4.2 的外迭代步数是逐渐减小的, 但是减小的趋势是下降的. 而对于算法 3.1 和 4.1, 外迭代步数受内迭代步数增加的敏感性并不强.

表 3 求解融合 Lasso 问题 (1.4), 选取参数类型 I 时, 算法 3.1、3.2、4.1 和 4.2 所得数值结果

内迭代步数	方法	$\epsilon = 10^{-4}$			$\epsilon = 10^{-8}$		
		NMSD	SNR(dB)	Iter	NMSD	SNR(dB)	Iter
2	算法 3.1	0.0062	44.2197	385	0.0060	44.5044	500
	算法 3.2	0.0063	44.0661	510	0.0060	44.5044	659
	算法 4.1	0.0061	44.2254	385	0.0060	44.5044	500
	算法 4.2	0.0063	44.0643	510	0.0060	44.5044	659
10	算法 3.1	0.0062	44.1763	385	0.0060	44.5044	505
	算法 3.2	0.0062	44.2187	386	0.0060	44.5044	505
	算法 4.1	0.0062	44.1763	385	0.0060	44.5044	505
	算法 4.2	0.0062	44.2176	386	0.0060	44.5044	505
20	算法 3.1	0.0062	44.1723	385	0.0060	44.5044	506
	算法 3.2	0.0062	44.1762	385	0.0060	44.5044	505
	算法 4.1	0.0062	44.1720	385	0.0060	44.5044	506
	算法 4.2	0.0062	44.1752	385	0.0060	44.5044	505

表 4 求解融合 Lasso 问题 (1.4), 选择参数类型 II 时, 算法 3.1、3.2、4.1 和 4.2 所得数值结果

内迭代步数	方法	$\epsilon = 10^{-4}$			$\epsilon = 10^{-8}$		
		NMSD	SNR(dB)	Iter	NMSD	SNR(dB)	Iter
2	算法 3.1	0.0061	44.3020	386	0.0060	44.5044	506
	算法 3.2	0.0066	43.6520	683	0.0060	44.5044	895
	算法 4.1	0.0061	44.2980	386	0.0060	44.5044	500
	算法 4.2	0.0066	43.6464	683	0.0060	44.5044	895
10	算法 3.1	0.0062	44.1838	385	0.0060	44.5044	505
	算法 3.2	0.0062	44.1873	392	0.0060	44.5044	514
	算法 4.1	0.0062	44.1826	385	0.0060	44.5044	505
	算法 4.2	0.0062	44.1864	392	0.0060	44.5044	514
20	算法 3.1	0.0062	44.1756	385	0.0060	44.5044	506
	算法 3.2	0.0062	44.2183	386	0.0060	44.5044	506
	算法 4.1	0.0062	44.1747	385	0.0060	44.5044	506
	算法 4.2	0.0062	44.2179	386	0.0060	44.5044	506

进一步, 当选取内迭代步数为 10 和参数类型 I 时, 图 4 展示了目标函数值和 SNR 值随着迭代步数增加的变化曲线. 我们能从图 4 看到 4 种迭代算法的相应变化曲线几乎完全重合.

5.2 约束全变分正则化问题

本小节运用所提出迭代算法求解具有约束的全变分正则化问题 (1.5). 特别地, 我们将模型应用于不完备投影下的计算机断层成像 (computed tomography, CT) 重建问题. 首先, 简要回顾离散的 TV 范数定义.

定义 5.1 ^[27] 设 $x \in \mathbb{R}^n$ 是一个 $\sqrt{n} \times \sqrt{n}$ 图像按其列的顺序拉成的列向量. 记 $\|\cdot\|_1$ 表示通常的

图 4 求解融合 Lasso 问题 (1.4), 内迭代步数为 10 和参数类型 I 时, (a) 目标函数值和迭代数的关系图; (b) 信噪比 (SNR) 值和迭代数的关系图

1- 范数. 首先, 定义一阶差分矩阵 B 如下:

$$B_{n \times n} = \begin{pmatrix} -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix},$$

和矩阵

$$D = \begin{pmatrix} I \otimes B \\ B \otimes I \end{pmatrix}, \quad (5.2)$$

其中 $\otimes$ 表示 Kronecker 内积, I 表示单位矩阵.

(i) 各项同性 TV 范数 (isotropic total variation, ITV), 定义是 $\|x\|_{\text{ITV}} = \|Dx\|_{2,1}$, 其中

$$\|y\|_{2,1} = \sum_{i=1}^n \sqrt{y_i^2 + y_{n+i}^2}, \quad y \in \mathbb{R}^{2n \times 1};$$

(ii) 各项异性 TV 范数 (anisotropic total variation, ATV), 定义是 $\|x\|_{\text{ATV}} = \|Dx\|_1$.

注意到具有约束的全变分正则化问题 (1.5) 也是所考虑优化问题 (1.3) 的一种特殊情形. 事实上, 我们可以设 $f(x) = \frac{1}{2} \|Ax - b\|_2^2$, $g(x) = \delta_C(x)$ ($\delta_C(x)$ 表示定义在闭凸集上 C 的指示函数), $h(Bx) = \mu \|x\|_{\text{TV}}$. 考虑到重建图像的先验信息, 在本实验中, 取闭凸集 C 为非负集合, 即 $C = \{x \in \mathbb{R}^n \mid x_i \geq 0\}$. 我们使用标准的 256×256 Shepp-Logan 体模作为待重建的图像 (见图 5). 我们使用平行束的方式扫描图像, 在 0 到 360 的范围内选取 20 个投影角度, 每个投影角度下的投影射线束为 320 条射线. 实验仿真数据主要基于迭代重建工具箱 AIRtools^[63] 得到. 此时, 系统矩阵 A 是 $6,400 \times 65,536$. 我们对投影数据向量 b 施加以均值为 0、方差为 0.01 的随机 Gauss 噪声. 关于迭代参数的选取, 对于算法 3.1 和 4.1, 取 $\lambda = 1/\lambda_{\max}(DD^T)$. 在算法 3.2 和 4.2 中, 取 $\sigma = 1/\lambda_{\max}(DD^T)$, $\tau = 1$. 已知离散梯度矩阵 D (5.2), 有 $\lambda_{\max}(DD^T) = 8$. 选取正则参数 $\mu = 0.5$. 所得数值结果见表 5.

图 5 标准的 256×256 Shepp-Logan 模型图像

表 5 求解约束全变分正则化问题 (1.5), 算法 3.1、3.2、4.1 和 4.2 所得数值结果

内迭代步数	方法	$\epsilon = 10^{-4}$			$\epsilon = 10^{-6}$			$\epsilon = 10^{-8}$		
		NMSD	SNR(dB)	Iter	NMSD	SNR(dB)	Iter	NMSD	SNR(dB)	Iter
1	算法 3.1	0.1756	15.1103	1,036	0.0213	33.4403	9,524	0.0209	33.6169	14,615
	算法 3.2	0.2344	12.6014	1,021	0.0219	33.2001	16,799	0.0209	33.6160	27,983
	算法 4.1	0.1756	15.1102	1,036	0.0213	33.4404	9,523	0.0209	33.6169	14,615
	算法 4.2	0.2344	12.6014	1,021	0.0219	33.2002	16,799	0.0209	33.6160	27,982
2	算法 3.1	0.1756	15.1071	1,035	0.0213	33.4399	9,535	0.0209	33.6169	14,635
	算法 3.2	0.2035	13.8278	1,004	0.0217	33.2875	11,600	0.0209	33.6166	19,156
	算法 4.1	0.1756	15.1071	1,035	0.0213	33.4399	9,534	0.0209	33.6169	14,634
	算法 4.2	0.2035	13.8278	1,004	0.0217	33.2874	11,599	0.0209	33.6166	19,156
10	算法 3.1	0.1757	15.1038	1,034	0.0213	33.4396	9,545	0.0209	33.6169	14,652
	算法 3.2	0.1758	15.0997	1,034	0.0213	33.4388	9,548	0.0209	33.6169	14,665
	算法 4.1	0.1757	15.1038	1,034	0.0213	33.4397	9,545	0.0209	33.6169	14,652
	算法 4.2	0.1758	15.0997	1,034	0.0213	33.4388	9,548	0.0209	33.6169	14,665

我们能从表 5 中看出, 迭代算法 3.1 和 4.1 具有一致的数值结果. 这个结果表明算法 3.1 和 4.1 之间的某种等价性. 类似地, 在给定内迭代步数时, 算法 3.2 和 4.2 的数值结果同样基本相同. 注意到, 当内迭代步数为 1 时, 基于原始对偶方法的算法 3.2 和 4.2 所需迭代步数多于根据对偶方法推导出的算法 3.1 和 4.1. 通过增加内迭代步数, 为了达到相同的误差精度, 算法 3.1、3.2、4.1 和 4.2 所需外迭代步数越来越接近. 当内迭代步数为 10 和 $\epsilon = 10^{-8}$ 时, 所有迭代算法所需迭代步数几乎完全相等. 相应的目标函数值和 SNR 值随着迭代步数增加的变化见图 6. 相应重建所得图像见图 7.

5.3 低秩全变分图像超分辨率重建问题

在各种医学成像问题中, 如 CT、MRI 和 PET 等, 对高分辨率的图像有巨大的需求. 然而, 在实际中, 由于成像系统物理限制条件等因素, 往往得到的是较低分辨率的图像. 因此, 如何从低分辨率的图像重构出高分辨率的图像具有非常重要的意义和价值. 超分辨率重建就是为解决问题而提出的. 本节应用所提的迭代算法求解低秩全变分 (LRTV) 图像超分辨率重建优化问题 (1.7). 在 LRTV 优化问题 (1.7) 中, 包含两个正则项: 一个是核范数 $\|X\|_*$, 它是低秩约束的凸松弛; 另一个是 TV 范数

图 6 求解约束全变分正则化问题 (1.5), 当内迭代步数为 10 和 $\epsilon = 10^{-8}$ 时, (a) 目标函数值和迭代数的关系图; (b) 信噪比 (SNR) 值和迭代数的关系图

图 7 求解约束全变分正则化问题 (1.5), 当内迭代步数为 10 和 $\epsilon = 10^{-8}$ 时, 迭代算法重建所得图像

$\|X\|_{\text{TV}}$, 它能够表示为一个凸函数与线性算子的复合形式, 于是可以看到 LRTV 优化问题 (1.7) 是优化问题 (1.3) 的一个特殊情形. 设 $f(X) = \frac{1}{2}\|DSX - T\|_F^2$, $g(X) = \lambda_1\|X\|_*$, $h(BX) = \lambda_2\|X\|_{\text{TV}}$. 值得注意的是, 核范数的邻近算子具有一个显示封闭形式解, 具体可参见文献 [64].

在本实验中, 我们使用与文献 [51] 相同的数据集. 具体来说, 选取来自 Brainweb¹⁾ 的 T1 MR phantom 作为标准, 选择大小 217×181 分辨率为 1 mm 的 2D 图像为测试图像, 见图 8.

1) [Http://www.bic.mni.mcgill.ca/brainweb](http://www.bic.mni.mcgill.ca/brainweb)

图 8 原始高分辨率图像

模糊算子 S 被定义为方差为 1 的 Gauss 核. 下采样算子是计算模糊图像 4 个像素的平均值. 上采样算子通过最近邻插值定义. 为了评估重建图像的质量, 除了 SNR 和 NMSD 外, 我们也使用结构相似性指数 (structural similarity index, SSIM)^[65] 去评估重建图像的质量. SSIM 的定义为

$$\text{SSIM}(f, g) = \frac{(2\mu_f\mu_g + c_1)(2\sigma_{fg} + c_2)}{(\mu_f^2 + \mu_g^2 + c_1)(\sigma_f^2 + \sigma_g^2 + c_2)},$$

其中 μ_f 是原始高分辨率图像 f 的平均值, μ_g 是修复后的图像 g 的平均值, σ_f^2 和 σ_g^2 分别是 f 和 g 的方差, σ_{fg} 是 f 和 g 的协方差, $c_1 = (k_1 L)^2$, $c_2 = (k_2 L)^2$, $k_1 = 0.01$, $k_2 = 0.03$, L 是像素值的动态范围. SSIM 的范围是 0 到 1, 当两幅图一模一样时, SSIM 的值等于 1.

为测试迭代算法的重建效果, 根据文献 [51], 分别选取正则参数 $\lambda_1 = 0.01$ 和 $\lambda_2 = 0.01$. 在这个数值实验中, 由于我们不知道下采样算子 D 和模糊算子 S 的精确算子范数值, 因此, 我们通过调节参数 γ 尽可能地确保所提迭代算法收敛. 最终取 $\gamma = 0.1$, 其他的迭代参数选择与第 5.2 小节相同. 所得数值结果见表 6. 符号 “—” 表示迭代算法超过设定最大迭代步数 1×10^5 . 我们能看到当停止标准 $\epsilon = 10^{-8}$ 和内迭代步数为 10 时, 所提的迭代算法对于给定评价指标 NMSD、SNR 和 SSIM 几乎完全相同. 相应地, 在图 9 中, 我们给出 SNR 值和目标函数值随着迭代步数增加的变化曲线. 我们能从图 9 看出 SNR 值并不是严格单调增加的, 而是开始增加快后面缓慢减小. 对于所有提出的迭代算法, SNR

表 6 求解低秩全变分图像超分辨率重建问题 (1.7), 算法 3.1、3.2、4.1 和 4.2 所得数值结果

内迭代步数	方法	$\epsilon = 10^{-6}$				$\epsilon = 10^{-8}$			
		NMSD	SNR(dB)	SSIM	Iter	NMSD	SNR(dB)	SSIM	Iter
1	算法 3.1	0.0781	22.1440	0.9796	34,573	0.0797	21.9701	0.9642	—
	算法 3.2	0.0890	21.0146	0.9762	19,046	0.0773	22.2393	0.9635	—
	算法 4.1	0.0781	22.1440	0.9796	34,573	0.0797	21.9701	0.9642	—
	算法 4.2	0.0890	21.0146	0.9762	19,046	0.0773	22.2393	0.9635	—
2	算法 3.1	0.0781	22.1440	0.9602	34,573	0.0797	21.9701	0.9642	—
	算法 3.2	0.0814	21.7906	0.9549	29,252	0.0783	22.1244	0.9650	—
	算法 4.1	0.0781	22.1440	0.9602	34,573	0.0797	21.9701	0.9642	—
	算法 4.2	0.0814	21.7906	0.9549	29,252	0.0783	22.1244	0.9650	—
10	算法 3.1	0.0781	22.1440	0.9602	34,573	0.0797	21.9701	0.9642	—
	算法 3.2	0.0781	22.1435	0.9602	34,576	0.0797	21.9707	0.9642	—
	算法 4.1	0.0781	22.1440	0.9602	34,573	0.0797	21.9701	0.9642	—
	算法 4.2	0.0781	22.1435	0.9602	34,576	0.0797	21.9707	0.9642	—

图 9 求解低秩全变分图像超分辨率重建问题 (1.7), 当 $\epsilon = 10^{-8}$ 和内迭代步数为 10 时, (a) 目标函数值和迭代数的关系图; (b) 信噪比 (SNR) 值和迭代数的关系图

图 10 求解低秩全变分图像超分辨率重建问题 (1.7), 当 $\epsilon = 10^{-8}$ 和内迭代步数为 10 时, 迭代算法重建所得图像最大值是相同的且等于 22.2399 (dB). 相对应的迭代步数分别是 48,671、48,718、48,671 和 48,718. 这个结果与我们前面的数值结果基本一致. 为了直观展现重建图像, 图 10 展示了所提出的迭代算法当停止标准 $\epsilon = 10^{-8}$ 和内迭代步数为 10 时得到的重建图像.

6 小结

本文研究了一类三个凸函数和的凸优化问题 (1.3). 为了得到有效的迭代算法来解决所考虑的优化问题 (1.3), 我们在充分利用了目标函数的可微凸函数的性质基础上, 采用了两种算子分裂算法: 向前向后分裂算法和三算子分裂算法. 在两种迭代算法中, 我们分别计算了函数 $g + h \circ B$ 和 $h \circ B$ 的邻近算子. 考虑到这些邻近算子一般都没有封闭形式的显示解, 我们分别运用对偶和原始对偶方法的思想设计了有效的求解迭代算法. 通过与现有其他算法比较, 我们发现用于求解优化问题 (1.3) 的三个现有迭代算法与我们提出的迭代算法之间的联系. 同时, 我们证明了所提出的迭代算法在有限维 Hilbert 空间中的收敛性. 特别地, 我们还提供了一种分解方法来选择 Condat-Vu 算法的迭代参数. 最后, 我

们将所提迭代算法分别应用于融合 Lasso 问题 (1.4)、约束全变分正则化问题 (1.5) 和低秩全变分图像超分辨率重建问题 (1.7), 所得数值实验结果验证了我们提出的迭代算法的有效性. 对于内迭代的步数选择问题, 我们的数值结果证实它可以选择一较小的常数足以保证迭代算法的收敛. 在未来的工作中, 我们将考虑通过使用线搜索等方法来改进所提出的迭代算法, 而不需要估计 ∇f 的 Lipschitz 常数以及算子 B 的范数.

致谢 感谢审稿人对本文提出的宝贵意见, 根据他们的修改意见, 使得本文的质量得以极大提升. 同时也感谢编委在处理本文时所付出的辛勤劳动.

参考文献

- 1 Combettes P L, Wajs V R. Signal recovery by proximal forward-backward splitting. *Multiscale Model Simul*, 2005, 4: 1168–1200
- 2 Daubechies I, Defrise M, De Mol C. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Comm Pure Appl Math*, 2004, 57: 1413–1457
- 3 Beck A, Teboulle M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J Imaging Sci*, 2009, 2: 183–202
- 4 Nesterov Y. A method for unconstrained convex minimizing problem with the rate of convergence $o(1/k^2)$. *Doklady AN SSSR*, 1983, 269: 543–547
- 5 Zeng J S, Fang J, Xu Z B. Sparse SAR imaging based on $L_{1/2}$ regularization. *Sci China Inf Sci*, 2012, 55: 1755–1775
- 6 曾锦山, 何涛, 欧阳诗康. 稀疏正则化: Gauss-Seidel 阈值迭代算法. *中国科学: 数学*, 2018, 48: 969–986
- 7 Candes E J, Romberg J, Tao T. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Trans Inform Theory*, 2006, 52: 489–509
- 8 Rudin L I, Osher S, Fatemi E. Nonlinear total variation based noise removal algorithms. *Phys D*, 1992, 60: 259–268
- 9 Chan T F, Golub G H, Mulet P. A nonlinear primal-dual method for total variation-based image restoration. *SIAM J Sci Comput*, 1999, 20: 1964–1977
- 10 Bioucas-Dias J M, Figueiredo M A T. A new twist: Two-step iterative shrinkage/thresholding algorithms for image restoration. *IEEE Trans Image Process*, 2007, 16: 2992–3004
- 11 Zhu M Q, Wright S J, Chan T F. Duality-based algorithms for total-variation-regularized image restoration. *Comput Optim Appl*, 2010, 47: 377–400
- 12 Bonettini S, Ruggiero V. On the convergence of primal-dual hybrid gradient algorithms for total variation image restoration. *J Math Imaging Vision*, 2012, 44: 236–253
- 13 Zhang Y, Liu J, Yang W et al. Image super-resolution based on structure-modulated sparse representation. *IEEE Trans Image Process* 2015, 24: 2797–2810
- 14 Wen Y W, Chan R H, Zeng T Y. Primal-dual algorithms for total variation based image restoration under Poisson noise. *Sci China Math*, 2016, 59: 141–160
- 15 Beck A, Teboulle M. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *IEEE Trans Image Process*, 2009, 18: 2419–2434
- 16 Chambolle A. An algorithm for total variation minimization and applications. *J Math Imaging Vision*, 2004, 20: 89–97
- 17 Chambolle A, Pock T. A first-order primal-dual algorithm for convex problems with applications to imaging. *J Math Imaging Vision*, 2011, 40: 120–145
- 18 Zhu M, Chan T F. An efficient primal-dual hybrid gradient algorithm for total variation image restoration. *UCLA CAM Report cam08-34*, 2008
- 19 Esser E, Zhang X Q, Chan T F. A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science. *SIAM J Imaging Sci*, 2010, 3: 1015–1046
- 20 Zhang X, Burger M, Osher S. A unified primal-dual algorithm framework based on Bregman iteration. *J Sci Comput*, 2011, 46: 20–46
- 21 He B, Yuan X. Convergence analysis of primal-dual algorithms for a saddle-point problem: From contraction perspective. *SIAM J Imaging Sci*, 2012, 5: 119–149
- 22 He B, Ma F, Yuan X. An algorithmic framework of generalized primal-dual hybrid gradient methods for saddle point problems. *J Math Imaging Vision*, 2017, 58: 279–293
- 23 Yu Y, Peng J. A modified primal-dual method with applications to some sparse recovery problems. *Appl Math Comput*, 2018, 333: 76–94
- 24 Boyd S, Parikh N, Chu E, et al. Distributed optimization and statistical learning via the alternating direction method

- of multipliers. *Found Trends Mach Learn*, 2010, 3: 1–122
- 25 Esser E. Applications of Lagrangian-based alternating direction methods and connections to split Bregman. *UCLA CAM Report cam09-31*, 2009
 - 26 Chen P J, Huang J G, Zhang X Q. A primal-dual fixed point algorithm for convex separable minimization with applications to image restoration. *Inverse Problems*, 2013, 29: 025011
 - 27 Micchelli C A, Shen L, Xu Y. Proximity algorithms for image models: Denoising. *Inverse Problems*, 2011, 27: 045009
 - 28 Loris I, Verhoeven C. On a generalization of the iterative soft-thresholding algorithm for the case of non-separable penalty. *Inverse Problems*, 2011, 27: 125007
 - 29 Komodakis N, Pesquet J. Playing with duality: An overview of recent primal-dual approaches for solving large-scale optimization problems. *IEEE Signal Process Mag*, 2015, 32: 31–54
 - 30 Condat L. A primal-dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms. *J Optim Theory Appl*, 2013, 158: 460–479
 - 31 Vu B C. A splitting algorithm for dual monotone inclusions involving cocoercive operators. *Adv Comput Math*, 2013, 38: 667–681
 - 32 Combettes P L, Pesquet J C. Primal-dual splitting algorithm for solving inclusions with mixtures of composite, Lipschitzian, and parallel-sum type monotone operators. *Set-Valued Anal*, 2012, 20: 307–330
 - 33 Li Q, Zhang N. Fast proximity-gradient algorithms for structured convex optimization problems. *Appl Comput Harmon Anal*, 2016, 41: 491–517
 - 34 Wen M, Peng J, Tang Y, et al. A preconditioning technique for first-order primal-dual splitting method in convex optimization. *Math Probl Eng*, 2017, 2017: 3694525
 - 35 Pock T, Chambolle A. Diagonal preconditioning for first order primal-dual algorithms in convex optimization. In: *Proceedings of IEEE International Conference on Computer Vision (ICCV)*. Washington: IEEE, 2011, 1762–1769
 - 36 Latafat P, Patrinos P. Asymmetric forward-backward-adjoint splitting for solving monotone inclusions involving three operators. *Comput Optim Appl*, 2017, 68: 57–93
 - 37 Krol A, Li S, Shen L, et al. Preconditioned alternating projection algorithms for maximum a posteriori ECT reconstruction. *Inverse Problems*, 2012, 28: 115005
 - 38 Chen P, Huang J, Zhang X. A primal-dual fixed point algorithm for minimization of the sum of three convex separable functions. *Fixed Point Theory Appl*, 2016, 2016: 54
 - 39 Yan M. A new primal-dual algorithm for minimizing the sum of three functions with a linear operator. *J Sci Comput*, 2018, 76: 1698–1717
 - 40 Davis D, Yin W. A three-operator splitting scheme and its optimization applications. *Set-Valued Var Anal*, 2017, 25: 829–858
 - 41 Tibshirani R, Saunders M, Rosset S, et al. Sparsity and smoothness via the fused Lasso. *J R Stat Soc Ser B Stat Methodol*, 2005, 67: 91–108
 - 42 Tibshirani R, Wang P. Spatial smoothing and hot spot detection for CGH data using the fused Lasso. *Biostatistics*, 2008, 9: 18–29
 - 43 Liu J, Yuan L, Ye J. An efficient algorithm for a class of fused Lasso problems. In: *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York: ACM, 2010, 323–332
 - 44 Ye G B, Xie X H. Split Bregman method for large scale fused Lasso. *Comput Statist Data Anal*, 2011, 55: 1552–1569
 - 45 Li X, Mo L, Yuan X, et al. Linearized alternating direction method of multipliers for sparse group and fused LASSO models. *Comput Statist Data Anal*, 2014, 79: 203–221
 - 46 Friedman J, Hastie T, Höfling H, et al. Pathwise coordinate optimization. *Ann Appl Stat*, 2007, 1: 302–332
 - 47 Shi H, Tu S, Xu Y, et al. A primer on coordinate descent algorithms. *UCLA CAM Report cam16-67*, 2016
 - 48 Sidky E Y, Pan X C. Image reconstruction in circular cone-beam computed tomography by constrained, total-variation minimization. *Phys Med Biol*, 2008, 53: 4777–4807
 - 49 Hager W, Ngo C, Yashtini M, et al. An alternating direction approximate Newton algorithm for ill-conditioned inverse problems with application to parallel MRI. *J Oper Res Soc China*, 2015, 3: 139–162
 - 50 Shi F, Cheng J, Wang L, et al. Low-rank total variation for image super-resolution. In: *Proceedings of Medical Image Computing and Computer-Assisted Intervention-MICCAI*. New York: Springer, 2013, 155–162
 - 51 Shi F, Cheng J, Wang L, et al. LRTV: MR image super-resolution with low-rank and total variation regularizations. *IEEE Trans Med Imaging*, 2015, 34: 2459–2466
 - 52 Marquina A, Osher S J. Image super-resolution by TV-regularization and Bregman iteration. *J Sci Comput*, 2008, 37: 367–382
 - 53 Zalinescu C. *Convex Analysis in General Vector Spaces*. Singapore: World Scientific, 2002
 - 54 Moreau J. Fonctions convexes duales et points proximaux dans un espace hilbertien. *C R Acad Sci Paris Ser A Math*,

- 1962, 255: 2897–2899
- 55 Parikh N, Boyd S. Proximal algorithms. *Found Trends Optim*, 2014, 1: 123–231
 - 56 Bauschke H, Combettes P. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. London: Springer, 2011
 - 57 Combettes P L, Dung D, Vu B C. Dualization of signal recovery problems. *Set-Valued Anal*, 2010, 18: 373–404
 - 58 Combettes P L. Solving monotone inclusions via compositions of nonexpansive averaged operators. *Optimization*, 2004, 53: 475–504
 - 59 Lions P L, Mercier B. Splitting algorithms for the sum of two nonlinear operators. *SIAM J Numer Anal*, 1979, 16: 964–979
 - 60 Eckstein J, Bertsekas D P. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Math Program*, 1992, 55: 293–318
 - 61 He B, Liu H, Wang Z, et al. A strictly contractive Peaceman-Rachford splitting method for convex programming. *SIAM J Optim*, 2014, 24: 1011–1040
 - 62 Briceño-Arias L M. Forward-Douglas-Rachford splitting and forward-partial inverse method for solving monotone inclusions. *Optimization*, 2015, 64: 1239–1261
 - 63 Hansen P C, Saxild-Hansen M. AIR tools—a MATLAB package of algebraic iterative reconstruction methods. *J Comput Appl Math*, 2012, 236: 2167–2178
 - 64 Cai J F, Candès E J, Shen Z W. A singular value thresholding algorithm for matrix completion. *SIAM J Optim*, 2010, 20: 1956–1982
 - 65 Wang Z, Bovik A C, Sheikh H R, et al. Image quality assessment: From error visibility to structural similarity. *IEEE Trans Image Process*, 2004, 13: 600–612

An inner-outer iteration method for solving convex optimization problems involving the sum of three convex functions

Yuchao Tang, Guorong Wu & Chuanxi Zhu

Abstract In recent years, the optimization problem of the sum of several convex functions has received much attention. In this paper, we consider solving a class of convex optimization problem which minimizes the sum of three convex functions $f(x) + g(x) + h(Bx)$, where $f(x)$ is differentiable with a Lipschitz continuous gradient, $g(x)$ and $h(x)$ have a closed-form expression of their proximity operators and B is a bounded linear operator. Such optimization problems have wide application in signal recovery and image processing. To make full use of the differentiable function in the problem, we propose several inner-outer iterative algorithms based on the forward-backward splitting algorithm and the three-operator splitting algorithm frameworks. In the process of deriving the iterative algorithms, we use dual and primal-dual methods to solve the proximity operator of the functions $g + h \circ B$ and $h \circ B$. Under mild assumptions on the parameters, we prove the convergence of the proposed iterative algorithms. By comparing with the Condat and Vu algorithm, the primal-dual fixed point (PDFP) algorithm and the primal-dual three-operator (PD3O) algorithm, we establish the connection between these algorithms with ours. Numerical experiments applied to the fused Lasso problem, the constrained total variation regularization problem and the low-rank total variation image super-resolution problem demonstrate the effectiveness and efficiency of the proposed iterative algorithms.

Keywords forward-backward splitting algorithm, three operator splitting algorithm, dual, primal-dual, total variation

MSC(2010) 90C25, 65K10

doi: 10.1360/SCM-2017-0313