

针对分式 l_1 正则化优化问题的自适应ADMM算法

袁海洋¹, 廖春美^{2*}, 李明华²

¹重庆交通大学数学与统计学院, 重庆

²重庆文理学院数学与人工智能学院, 重庆

收稿日期: 2025年11月17日; 录用日期: 2025年12月10日; 发布日期: 2025年12月23日

摘要

本文研究了一种基于分式 l_1 正则化的非凸优化模型, 该模型能比传统 l_1 范数更精确地逼近 l_0 范数的稀疏性。为高效求解此非凸优化问题, 我们设计了基于交替方向乘子法的求解框架, 并提出了一种自适应的惩罚参数与正则化参数调整策略。该策略能动态调整惩罚参数以平衡原始残差与对偶残差, 加速收敛; 同时, 能自适应地调整正则化参数以维持数据拟合项与正则项的平衡, 有效引导迭代过程趋向高质量解。数值实验表明, 与现有快速算法相比, 本文的方法在求解精度上具有明显的优势, 能够稳定地获得更低相对误差的解。而且, 该自适应策略增强了对超参数初始设置的鲁棒性, 使算法在宽松的参数条件下仍能保持较好的恢复性能, 为解决分式 l_1 正则化优化问题提供了一个高精度且实用的解决方案。

关键词

稀疏信号恢复, 分式 l_1 正则化, 交替方向乘子法, 自适应参数

Adaptive ADMM for Optimization Problems with Fractional l_1 Regularization

Haiyang Yuan¹, Chunmei Liao^{2*}, Minghua Li²

¹School of Mathematics and Statistics, Chongqing Jiaotong University, Chongqing

²School of Mathematics and Artificial Intelligence, Chongqing University of Arts and Sciences, Chongqing

Received: November 17, 2025; accepted: December 10, 2025; published: December 23, 2025

Abstract

This paper investigates a nonconvex optimization model based on fractional l_1 regularization,

*通讯作者。

which provides a superior approximation to the sparsity of the l_0 norm compared to conventional l_1 norm. To address this, our solution is built around the Alternating Direction Method of Multipliers (ADMM) framework, augmented with a novel strategy for the adaptive tuning of both penalty and regularization parameters. This strategy dynamically balances the primal and dual residuals to accelerate convergence, while adaptively maintaining the equilibrium between the data fidelity and regularization terms, thereby steering the iterations toward high-quality solutions. Our numerical tests show that the proposed method significantly outperforms existing fast algorithms in solution accuracy, reliably achieving lower relative errors. It also exhibits greater robustness, as the adaptive scheme reduces sensitivity to initial hyperparameter choices, allowing it to perform well even with less stringent parameter tuning. Consequently, our approach provides a highly practical and precise solver for problems involving fractional regularization.

Keywords

Sparse Signal Recovery, Fractional l_1 Regularization, Alternating Direction Method of Multipliers, Adaptive Parameters

Copyright © 2025 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

稀疏信号恢复是现代信号处理、统计学与机器学习中的核心任务之一，其目标是从远少于信号维度的线性观测中精确恢复原始信号。该问题的一般模型是在线性模型的拟合误差上，增加一个用于诱导稀疏性的正则化项。其中， l_1 范数正则化因其凸性和求解便利而在信号与图像处理，计算机视觉，生物信息学等领域被广泛应用[1]-[3]。然而，有理论与实践表明， l_1 范数会对大值系数产生过度惩罚，使得恢复信号产生偏差，影响其在复杂场景下的最终精度[4]。

研究者们为了克服 l_1 范数的这种局限性，提出了用非凸正则化函数来替代传统的 l_1 范数。而且已有研究表明，使用非凸正则化函数逼近 l_0 范数可以得到比 l_1 范数更稀疏的解。分式 l_1 正则化模型正是其中一种具有代表性的模型，它通过超参数 r ，构建了分数形式的正则化函数，该函数在 $r \rightarrow \infty$ 时无限趋近于理想的 l_0 范数。本文主要研究如下形式的分式 l_1 正则化优化问题：

$$\min_{x \in \mathbb{R}^n} F(x) := \frac{1}{2} \|Ax - b\|_2^2 + \lambda \sum_{i=1}^n \frac{r|x_i|}{1+r|x_i|}. \quad (1)$$

其中 b 为观测向量， $A \in \mathbb{R}^{m \times n}$ 是感知矩阵($m \ll n$)， $x \in \mathbb{R}^n$ 是原始信号， $\lambda > 0$ 为正则化参数， x_i 为 x 的第 i 个分量， $i = 1, 2, \dots, n$ ， $r > 0$ 为超参数。

近年来，关于该问题的研究逐步深入。2016 年，崔安刚[5]提出了 FP-DC 算法，通过差分算法将分式 l_1 正则化问题转化为一系列凸子问题求解，用实验证明了 FP-DC 算法恢复成功率远优于 Soft 算法[6]和 Half 算法[7]。2019 年，Yazdanpanah 等人[8]用分式 l_1 正则项替换掉 l_0 范数，设计出了自适应沃尔泰拉滤波器，并用实验证明了该滤波器比经典方法有更好的性能。2020 年，李海洋等人[9]推导出了该模型的闭式阈值表达式，并据此提出了迭代 FP 阈值算法(FPTA)，实验结果表明，该算法在有噪声和无噪声的情况下均表现出了较好的恢复性能。2024 年，崔安刚等人[10]进一步引入动量加速技术，提出了快速 FP 算法，在保持高精度的同时减少了运行时间，实现了性能的进一步优化。

因为分式函数的非凸性与非光滑性, 想要对分式 l_1 正则化优化问题进行高效、鲁棒的数值求解会有较大困难。现有的一些算法在求解问题(1)时, 往往依赖于复杂的求解或嵌套迭代过程, 这不仅计算开销较大, 也使得算法整体对参数设置较为敏感, 在多样化的场景下不能保持稳定的性能。

而交替方向乘子法(ADMM)作为一类广泛应用于求解优化问题的算法, 早已被用于求解稀疏优化问题, 且能得到较好的结果[11]。本文的主要工作正是围绕 ADMM 框架进行, 在现有的研究基础上, 设计了一种针对分式 l_1 正则化优化问题的高精度的数值求解方案。

本文的主要贡献如下: 首先, 我们使用 ADMM 算法将该优化问题分成两个子问题, 并对子问题进行分析, 设计了一种向量化的数值求解策略。该策略通过变量变换与并行计算, 能高效地处理子问题, 避免传统求解中复杂的标量循环与条件判断, 显著提升了单次迭代的计算效率。其次, 我们提出了一种双参数自适应策略来解决惩罚参数和正则化参数的选择问题。该策略通过自动调整参数来维持原始残差与对偶残差之间、拟合项与正则项之间的平衡, 从而无需手动调参, 增强了算法的鲁棒性。最后, 通过系统性的数值实验, 我们将所提出的算法与主流稀疏恢复算法进行了对比。实验结果表明, 我们的方法在成功恢复稀疏信号时, 能得到更低的相对误差和更精确的解。

本文的后续安排如下: 第 2 节将介绍分式 l_1 正则化问题与 ADMM 求解框架, 详细阐述我们提出的带自适应机制的 ADMM 算法; 第 3 节通过可靠的数值实验结果展示算法的性能; 第 4 节对全文进行总结与展望。

2. 带自适应机制的精确 ADMM 算法

在本节中, 我们提出了一种求解基于分式函数的 l_1 正则化优化问题的带自适应参数机制的精确 ADMM 算法。这里求出了标准 ADMM 算法中的子问题的解析解, 并借鉴文献[12]中的自适应机制, 将其扩展到标准 ADMM 算法, 形成了自适应 ADMM 算法。

2.1. 标准 ADMM 算法

为了使用 ADMM 去解决问题(1), 我们首先引入了一个辅助变量 z 去将问题(1)转化为如下的带约束的等价形式:

$$\min_{x,z} \frac{1}{2} \|Ax - b\|_2^2 + \lambda \sum_{i=1}^n \frac{r|z_i|}{r|z_i| + 1} \quad \text{s.t.} \quad x = z. \quad (2)$$

问题(2)的增广拉格朗日函数为:

$$\mathcal{L}_\rho(x, z, y) = \frac{1}{2} \|Ax - b\|_2^2 + \lambda \sum_{i=1}^n \frac{r|z_i|}{1 + r|z_i|} + y^\top (x - z) + \frac{\rho}{2} \|x - z\|_2^2.$$

其中 $y \in \mathbb{R}^n$ 是拉格朗日乘子向量, $\rho > 0$ 是惩罚参数。ADMM 算法通过交替优化 x 、 z 和更新乘子 y 来求解, 其迭代步骤如下:

$$x^{k+1} = \arg \min_x \left\{ \frac{1}{2} \|Ax - b\|_2^2 + y^\top (x - z^k) + \frac{\rho}{2} \|x - z^k\|_2^2 \right\}; \quad (3)$$

$$z^{k+1} \in \arg \min_z \left\{ \lambda \sum_{i=1}^n \frac{r|z_i|}{1 + r|z_i|} + y^\top (x^{k+1} - z) + \frac{\rho}{2} \|x^{k+1} - z\|_2^2 \right\}; \quad (4)$$

$$y^{k+1} = y^k + \rho(x^{k+1} - z^{k+1}).$$

对于问题(3), 由于目标函数是二次凸函数, 其最小值点可以通过求梯度并将其设为零来得到唯一解

析解。其解析解为：

$$x^{k+1} = (A^\top A + \rho I)^{-1} \left(A^\top b + \rho \left(z^k - \frac{y}{\rho} \right) \right).$$

对于问题(4)，将与 z 相关的项合并，该子问题等价于：

$$z^{k+1} \in \arg \min_z \left\{ \lambda \sum_{i=1}^n \frac{r|z_i|}{1+r|z_i|} + \frac{\rho}{2} (z - v^{k+1})^2 \right\}.$$

其中 $v^{k+1} = x^{k+1} + \frac{y^k}{\rho}$ 。由于目标函数是可分离的，该问题可分解为 n 个独立的标量优化问题：

$$z_i^{k+1} \in \arg \min_{z_i} \left\{ \lambda r \frac{|z_i|}{1+r|z_i|} + \frac{\rho}{2} (z_i - v_i^{k+1})^2 \right\}, i = 1, 2, \dots, n. \quad (5)$$

这种可分离性是 ADMM 算法能够有效处理大规模问题的关键原因之一，因为它允许我们将一个复杂的高维优化问题分解为大量简单的、可并行处理的一维优化问题。

对于问题(5)易得：取得最优解时， z 的分量 z_i^{k+1} 的符号和 v_i^{k+1} 的符号相同：如果 $v_i^{k+1} \geq 0$ ，则 $z_i^{k+1} \geq 0$ ；如果 $v_i^{k+1} < 0$ ，则 $z_i^{k+1} < 0$ 。

基于符号一致性，我们将问题(5)简化为：

$$t^* \in \arg \min_{t \geq 0} \left\{ \lambda r \frac{t}{1+rt} + \frac{\rho}{2} (t - |v_i^{k+1}|)^2 \right\}. \quad (6)$$

此时， $z_i^{k+1} = \text{sign}(v_i^{k+1}) \cdot t^*$ 。通过对该非凸函数进行深入分析，并利用其一阶最优性条件，我们可以推导出最优解可通过求解一个三次方程得到，然后将所求三次方程的所有正实根与 0 带入问题(5)，比较目标函数值，选择使目标函数值最小的解作为最优解。

具体地，我们在(6)的基础上，将其转化为一个在非负实数域上求解的一维优化问题：

$$\min_{t \geq 0} g(t) = \lambda r \frac{t}{1+rt} + \frac{\rho}{2} (t - |v_i^{k+1}|)^2. \quad (7)$$

根据一阶最优性条件，令(7)的一阶导数为 0，有：

$$g'(t) = \lambda r \frac{1}{(1+rt)^2} + \rho (t - |v_i^{k+1}|) = 0. \quad (8)$$

在式(8)的基础上，整理可得：

$$\rho r^2 t^3 - \rho r (r|v_i^{k+1}| - 2) t^2 + \rho (1 - 2r|v_i^{k+1}|) t + (\lambda r - \rho |v_i^{k+1}|) = 0. \quad (9)$$

对式(9)两端乘上 r ，除以 ρ ，令 $s = rt$ ，化简得：

$$s^3 + (2 - r|v_i^{k+1}|)s^2 + (1 - 2r|v_i^{k+1}|)s + \left(\frac{\lambda r^2}{\rho} - r|v_i^{k+1}| \right) = 0. \quad (10)$$

问题(10)可以看作是一个关于 s 的一元三次方程，我们参考文献[9]，使用 Cardano 公式求解方程。然后将所求的所有正实根与 0 带入问题(7)，再比较 $g(t)$ 的值，选取使 $g(t)$ 值最小的解作为最优解。

在求解上面的 z 子问题时，我们为减少逐步求解分量的循环开销，采用了向量化的数值策略来求解所有分量。

2.2. 针对分式 L_1 正则化问题的自适应 ADMM 算法

在标准 ADMM 算法的基础上, 我们结合了自适应 ρ 策略和自适应 λ 策略, 提高了 ADMM 算法的收敛速度和解的精度。

2.2.1. 自适应 ρ 策略

首先, 我们定义原始残差 $r^k = \|x^k - z^k\|_2$ 和对偶残差 $s^k = \rho \|z^k - z^{k-1}\|_2$, 其中 r^k 用来衡量约束违反程度, s^k 用来衡量最优化条件的满足程度。

在 ADMM 算法中, 选择合适的惩罚参数 ρ 对收敛速度具有重要影响。当 ρ 值过大时, 虽然能够加强对等式约束 $x = z$ 的惩罚, 使原始残差快速下降, 但会导致 z 子问题中二次项系数过大, 使得 z 更新步长变小, 收敛变慢; 当 ρ 值过小时, 虽然 z 子问题更容易求解, 但会放松约束条件, 导致 x 和 z 差异过大, 使原始残差下降变慢。

为了能动态平衡原始残差和对偶残差的下降速度, 我们设计了自适应 ρ 调整策略。在每进行 10 次迭代时, 我们调整一次 ρ 值。如果 $r^k > \mu s^k$, 此时原始残差占主导, 约束违反程度较大, 通过将 ρ 更新为 $\tau \rho$ 来加强对等式约束的惩罚; 如果 $s^k > \mu r^k$, 此时对偶残差占主导, 说明 ρ 值可能过大导致子问题更新不稳定, 就将 ρ 更新为 $\tau^{-1} \rho$, 否则保持 ρ 不变。其中 $\mu > 1$, $\tau > 1$ 为调节参数。我们将 ρ 限制在一个合理区间内防止数值不稳定。

自适应 ρ 策略免除了手动调整 ρ 的过程, 实验结果表明, 采用自适应 ρ 策略后, 算法不仅和原来保持了相近的恢复精度, 还减少了约一半的运行时间。

2.2.2. 自适应 λ 策略

定义数据拟合项 $f_1 = \frac{1}{2} \|Ax - b\|_2^2$ 和稀疏正则项 $f_2 = \lambda \sum_{i=1}^n \frac{r|x_i|}{1+r|x_i|}$, 两项的比例为 $\text{ratio} = \frac{f_1}{f_2}$ 。

在目标函数中, 正则化参数 λ 在目标函数中起到平衡数据拟合项和稀疏正则项的重要作用。当 λ 过大时, 解过度稀疏, 可能丢失信号细节; 当 λ 过小时, 解不够稀疏, 拟合误差增大。我们提出了基于目标函数成分平衡的自适应 λ 策略: 如果 $\text{ratio} > 1 + \eta$, 说明拟合项远大于正则项, 为加强解的稀疏性, 将 λ 增大为 $\zeta \lambda$; 如果 $\text{ratio} < 1 + \eta$, 说明正则项占主导, 解可能过度稀疏而丢失细节, 通过将 λ 减小为 $\zeta^{-1} \lambda$ 来减弱稀疏性, 否则 λ 保持不变。其中 $\eta > 0$, $\zeta > 1$ 为调节参数, 同样, 我们会将 λ 限制在一定的范围内。该策略使得 λ 能够根据迭代过程动态调整, 从而引导算法趋向于一个拟合精度与稀疏性相平衡的解, 降低了对初始 λ 的依赖。

该自适应 λ 策略动态平衡了拟合项与正则项, 还不需要手动调整正则化参数 λ 。实验结果表明, 采用自适应 λ 策略后, 算法得到了比原来更精确的解。

结合了上述两种自适应策略, 显著增强了 ADMM 算法对问题的初始参数设置的鲁棒性。

为继续提高算法效率, 我们在 x 子问题求解过程中, 预计算了矩阵 $(A^\top A + \rho I)$, 并对其进行 Cholesky 分解, 再进行前向后向代换计算 x , 显著减少了计算量。然后充分利用 MATLAB 的向量运算优势, 将原来的按分量循环更新改为批量向量化计算, 一次性处理所有分量, 提高了算法效率。

在算法迭代过程中, 我们发现其产生的解分量往往很小但并非精确为零。为了得到真正的稀疏解, 我们在算法迭代收敛后采用了硬阈值处理。我们设定一个合理的正阈值 δ , 将所有绝对值小于 δ 的 x 分量精确置为零。具体的, 如果 $|x_i^{k+1}| < \delta$, 则令 $x_i^{k+1} = 0$, 否则 x_i^{k+1} 不变。其中阈值 δ 的选择尤为重要, 较大的 δ 可能会将有用的分量置为零, 导致解过于稀疏, 使恢复的误差增大, 而较小的 δ 无法将真正需要置零的分量置为零, 无法产生有效的稀疏解。实验证明, 在选取合适的 δ 后, 该方法不仅能在原来的算法上降低相对误差, 还能增强解的稀疏性。

结合以上内容, 我们给出针对问题(1)的带自适应策略的精确 ADMM 算法:

算法 1: 带自适应策略的精确 ADMM 算法

输入: A , \hat{x} , λ_0 , ρ_0 , r , b , tol , δ 。

初始化: 初始化 $x_0 = z_0 = y_0 = 0$, 设置 $k = 0$ 。

步骤 1:

$$\begin{aligned} x^{k+1} &= \arg \min_x \left\{ \frac{1}{2} \|Ax - b\|_2^2 + y^T(x - z^k) + \frac{\rho}{2} \|x - z^k\|_2^2 \right\}; \\ z^{k+1} &\in \arg \min_z \left\{ \lambda \sum_{i=1}^n \frac{r|z_i|}{1+r|z_i|} + y^T(x^{k+1} - z) + \frac{\rho}{2} \|x^{k+1} - z\|_2^2 \right\}; \\ v_i^{k+1} &= x_i^{k+1} + \frac{y_i^k}{\rho}; \\ t^* &\in \arg \min_{t \geq 0} \left\{ \lambda r \frac{t}{1+rt} + \frac{\rho}{2} (t - |v_i^{k+1}|)^2 \right\}; \\ z_i^{k+1} &= \text{sign}(v_i) \cdot t^*; \\ \text{更新 } \rho: \rho^{k+1} &= \begin{cases} \tau \rho^k & \text{若 } r^k > \mu s^k \\ \tau^{-1} \rho^k & \text{若 } s^k > \mu r^k; \\ \rho^k & \text{其他} \end{cases}; \\ \text{更新 } \lambda: \lambda^{k+1} &= \begin{cases} \zeta \lambda^k & \text{若 ratio} > 1 + \eta \\ \zeta^{-1} \lambda^k & \text{若 ratio} < 1 + \eta; \\ \lambda^k & \text{其他} \end{cases}; \\ y^{k+1} &= y^k + \rho(x^{k+1} - z^{k+1}); \end{aligned}$$

步骤 2: 检查收敛条件

如果不满足终止条件 $\max(r^k, s^k) \leq tol$ 或未达到最大迭代次数, 令 $k = k + 1$, 并转至步骤 1;

步骤 3: 如果 $|x_i^{k+1}| < \delta$, 则令 $x_i^{k+1} = 0$;

输出: $x^* = x^{k+1}$ 。

2.3. 计算复杂度分析

本节我们分析了算法 1 的计算复杂度, 其主要计算成本在于每次迭代中的求解 x 的问题(3)和求解 z 的问题(4)。

首先, 在求解问题(3)时, 我们对矩阵 $(A^T A + \rho I)$ 进行了预算算和 Cholesky 分解, 将其分解为下三角矩阵和其转置的乘积。该步骤有三重循环结构, 外层循环会遍历矩阵的每一列, 其计算复杂度为 $O(n)$, 中层循环会处理当前列的每个元素, 其平均计算复杂度为 $O(n)$, 内层循环会计算向量内积, 其平均计算复杂度也为 $O(n)$, 这种嵌套循环结构的总复杂度为 $O(n^3)$ 。分解之后, 我们利用下三角矩阵进行前向后向代换求解 x , 该步骤的平均计算复杂度为 $O(n^2)$ 。我们注意到, 当参数 ρ 改变时, 需要重新计算和分解矩阵 $(A^T A + \rho I)$, 此时计算复杂度为 $O(n^3)$ 。当参数 ρ 没有变化时, 不需要重新计算和分解, 在迭代过程中直接进行前向后向计算, 此时计算复杂度为 $O(n^2)$ 。所以求解问题(3)的整体的计算复杂度最小是 $O(n^2)$, 最大是 $O(n^3)$ 。

其次，在求解问题(4)时，我们要求解 n 个三次方程。每个三次方程都是通过 Cardano 公式求解，该步骤需要进行固定数量的算数运算，所以每个方程的求解复杂度为 $O(1)$ ，因此，求解 z 的问题(4)的总复杂度为 $O(n)$ 。

最后，算法 1 中的求解 y 的复杂度为 $O(n)$ ，计算残差的复杂度为 $O(n)$ ，调整参数 ρ 和参数 λ 的复杂度为 $O(1)$ 。所以算法 1 每次迭代的总复杂度最低为 $O(n^2)$ ，最高为 $O(n^3)$ 。

3. 数值实验

本章通过一系列数值实验，系统地评估了针对问题(1)所提出的自适应 ADMM 算法在稀疏信号恢复中的性能。所有实验均在 Intel Core i5-13500HX CPU，16GB 内存的个人计算机上使用 MATLAB R2019a 完成。

我们的实验核心聚焦于小规模的欠定系统，采用维度为 128×512 的感知矩阵 A ，其元素独立同分布于标准高斯分布，并进行了列归一化处理。真实的稀疏信号 \hat{x} 通过随机生成，其非零元位置均匀随机选取，非零数值服从标准正态分布。 s 表示稀疏度，即非零元素的个数。无噪声情况下的观测向量 b 由 $b = A\hat{x}$ 生成，因此，我们可以知道系统 $b = A\hat{x}$ 的真实稀疏解。为检验算法鲁棒性，我们也考虑了含噪场景，即 $b = A\hat{x} + \varepsilon$ ，其中 ε 为高斯白噪声，满足 $\varepsilon \in N(0, \sigma)$ 。

在后续的实验中，我们设置 r 为 0.5， δ 取 10^{-8} ，对于自适应策略的相关参数 μ 取 10， τ 取 1.5， η 取 0.4， ζ 取 1.5， λ_0 取 10^{-5} ， ρ_0 取 0.5，且 λ 和 ρ 的更新频率为 10，即每 10 次迭代判断一次是否需要更新 λ 和 ρ 。实验的停止准则设置为 $\max(s^k, r^k) \leq tol$ ，其中容差 tol 取 10^{-8} 。算法的性能主要通过相对误差 (Relative Error)、恢复成功率、运行时间 (Runtime) 来评判。相对误差表示为 $\|x^* - \hat{x}\|_2 / \|\hat{x}\|_2$ ，实验中使用 $\|x^* - \hat{x}\|_2^2 / \|\hat{x}\|_2^2$ 来衡量稀疏信号是否成功恢复 (设置其小于 10^{-5} 表示恢复成功)。

在第一个实验中，我们在稀疏信号的稀疏度为 10 的情况下，讨论了自适应策略的相关参数 μ ， τ ， η ， ζ 和更新频率的变化对自适应 ADMM 算法性能的影响。参数 μ 和参数 τ 是自适应 ρ 策略中的参数，参数 η 和参数 ζ 是自适应 λ 策略中的参数，我们分别画出了 $\mu - \tau$ 和 $\eta - \zeta$ 与相对误差和运行时间的热力图。如图 1 所示，子图 1(a) 为 $\mu - \tau$ 与相对误差的热力图，可以看到，当 τ 取 5 时，热力图颜色较亮，求出的最优解的相对误差明显较大。当 τ 取 1.3， μ 取 10 时，热力图颜色最深，相对误差最低。子图 1(b) 是 $\mu - \tau$ 与运行时间的热力图，可以看到 τ 在 1.3 和 1.5， μ 在 20 和 25 时，运行时间普遍较短。其中当 τ 取 1.3， μ 取 20 时运行时间最短。综合来看，取较小的 τ 会得到较好的相对误差和运行时间。子图 1(c) 是 $\mu - \tau$ 与相对误差的热力图，可以看到当 ζ 不变时， η 在一定范围内改变不会引起相对误差的变化，当 η 不变时， ζ 取 1.8 能使求出的最优解有最低的相对误差。子图 1(d) 是 $\eta - \zeta$ 与运行时间的热力图，可以看到当 η 不变时， ζ 取 1.8 能使运行时间最少，其中 η 取 0.6， ζ 取 1.8 能让运行时间最少。

图 2 是自适应参数更新频率与相对误差和运行时间的关系图。可以看到，更新频率在 10 到 390 的范围内，相对误差较为稳定，保持在 10^{-9} 这个量级，其中最低的相对误差在更新频率为 30 时取得。随着更新频率的增大，相对误差只在较小范围内波动。更新频率从 10 到 390 变化过程中，运行时间总体呈下降趋势，且在更新频率为 210 时，运行时间最低。若希望算法得到较高精度的解，可以选择较低的更新频率，例如在 30 左右。若希望算法运行时间短，则可以选择较高的更新频率，例如在 210 左右。若想平衡相对误差和运行时间，可以选择 150 左右的更新频率。

然后，我们测试了 FPTA 算法、标准 ADMM 算法和带自适应策略的 ADMM 算法在不同稀疏度的稀疏信号恢复中的成功率，对于每个稀疏度，我们重复 50 次实验，并给出平均结果。图 3 的左图显示了在稀疏度为 10 到 60 范围内，三个算法恢复稀疏信号的成功率。可以看到，在稀疏度在 10 到 35 时，三个算法都有极高的恢复成功率。在稀疏度为 40 左右时，带自适应策略的 ADMM 算法和 FPTA 算法恢复成

功率均高于标准的 ADMM 算法, FPTA 算法的恢复成功率略高于带自适应策略的 ADMM 算法, 但差距不大。图 3 的右图是三个算法的稀疏度和相对误差的关系图。可以很明显的看到, 在恢复成功的情况下, 带自适应策略的 ADMM 算法的相对误差的平方明显低于 FPTA 算法和标准 ADMM 算法。即与 FPTA 算法相比, 在信号恢复成功率差距不大的情况下, 我们的带自适应策略的 ADMM 算法能有更低的相对误差, 能得到更精确的解。

表 1 详细展示了无噪声的情况下标准 ADMM 算法、FPTA 算法和自适应 ADMM 算法在稀疏度为 10 到 20 间的相对误差(RE)和目标函数值(Fval)。True Fval 是由真实稀疏解 \hat{x} 计算得到的真实目标函数值, 显然, 由各个算法的恢复信号计算得到的目标函数值越接近真实目标函数值越好。在表 1 中, 标准 ADMM 算法的相对误差维持在 10^{-6} 左右的量级, FPTA 算法的相对误差稳定在 10^{-7} 左右的量级, 而带自适应策略的 ADMM 算法的相对误差稳定在 10^{-8} 左右的量级, 显著优于另外两种算法。同样, 对比另外两种算法, 带自适应的 ADMM 算法得到的目标函数值更加接近于真实的目标函数值。

表 2 详细展示了有噪声的情况下标准 ADMM 算法、FPTA 算法和自适应 ADMM 算法在稀疏度为 10 到 20 间的相对误差和目标函数值。 σ 取 10^{-5} 。在表 2 中, 标准 ADMM 算法的相对误差维持在 10^{-6} 左右的量级, FPTA 算法的相对误差和带自适应策略的 ADMM 算法的相对误差都稳定在 10^{-7} 左右的量级, 但带自适应策略的 ADMM 算法的相对误差略低于 FPTA 算法的相对误差。同样, 带自适应的 ADMM 算法得到的目标函数值更加接近于真实的目标函数值。

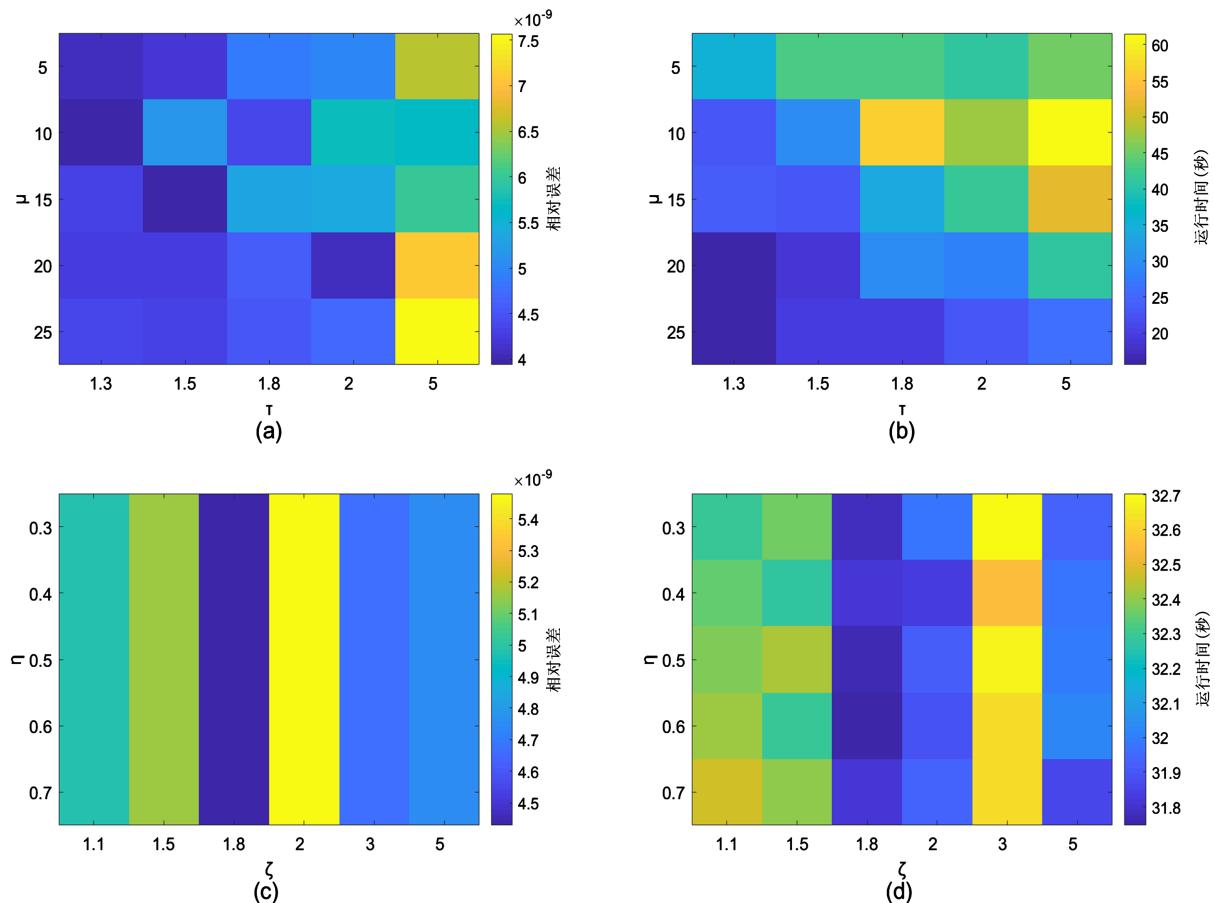


Figure 1. Heatmap of μ , τ , η , ζ versus relative error and runtime

图 1. μ , τ , η , ζ 与相对误差和运行时间的热力图

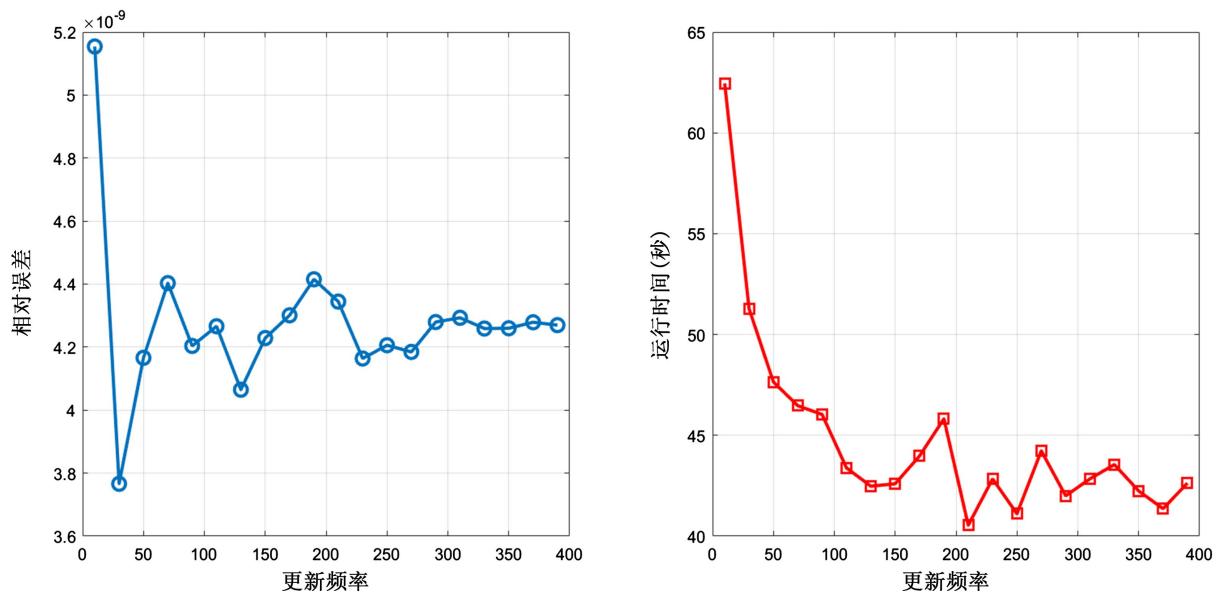


Figure 2. Relative error and runtime vs. update frequency
图 2. 更新频率与相对误差和运行时间的关系图

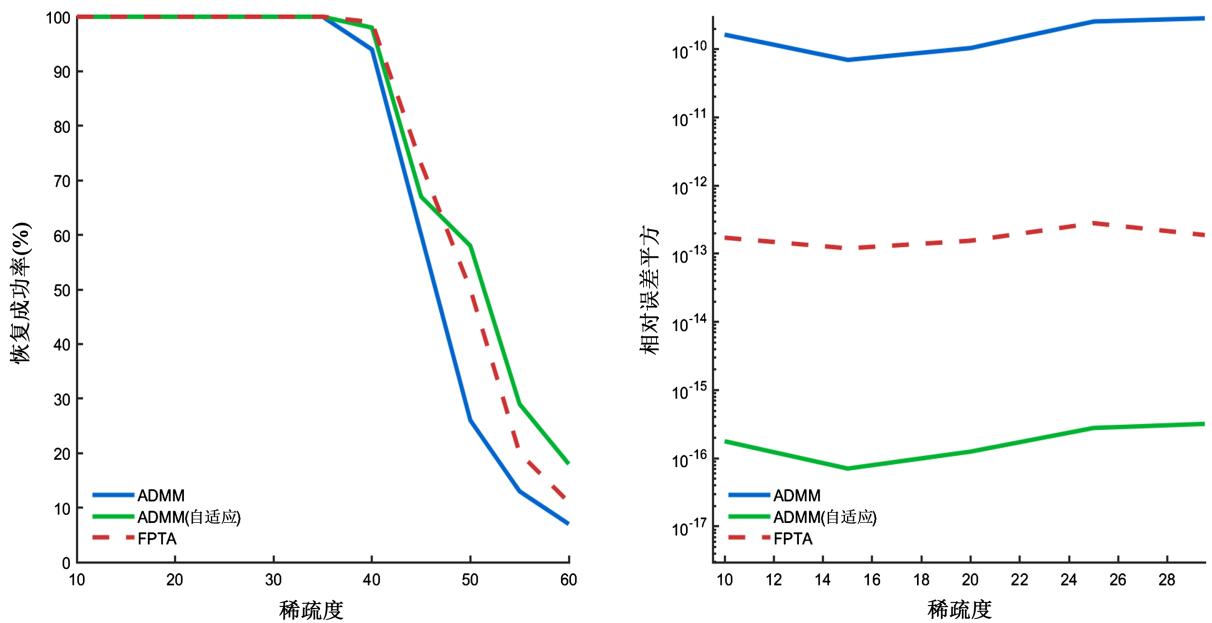


Figure 3. Recovery success rate and squared relative error vs. sparsity
图 3. 恢复成功率和相对误差平方与稀疏度的关系图

Table 1. Relative error and objective function value of each algorithm under noiseless conditions
表 1. 无噪声情况下各算法的相对误差和目标函数值

Noiseless	ADMM		ADMM (自适应)		FPTA			
	s	RE	Fval	RE	Fval	RE	Fval	True Fval
12	7.09e-06	3.55863737e-05	1.13e-08	3.55865619e-05	3.68e-07	3.55865557e-05	3.55865623e-05	
14	9.95e-06	4.36855718e-05	1.06e-08	4.36857762e-05	4.06e-07	4.36857662e-05	4.36857766e-05	
16	8.37e-06	4.44856287e-05	1.41e-08	4.44859469e-05	4.25e-07	4.44859351e-05	4.44859476e-05	

续表

18	1.69e-05	4.87488861e-05	1.77e-08	4.87492938e-05	3.92e-07	4.87492817e-05	4.87492947e-05
20	1.36e-05	5.39375803e-05	1.44e-08	5.39379883e-05	4.16e-07	5.39379718e-05	5.39379891e-05

Table 2. Relative error and objective function value of each algorithm under noisy conditions**表 2. 有噪声情况下各算法的相对误差和目标函数值**

Noisy	ADMM		ADMM(自适应)		FPTA		
	s	RE	Fval	RE	Fval	RE	Fval
12	8.07e-06	3.82232164e-05	3.23e-07	3.82233707e-05	3.68e-07	3.82233646e-05	3.82233690e-05
14	7.36e-06	4.96526113e-05	2.70e-07	4.96527865e-05	3.54e-07	4.96527716e-05	4.96527804e-05
16	8.71e-06	5.14598077e-05	2.73e-07	5.14600694e-05	2.76e-07	5.14600521e-05	5.14600611e-05
18	8.98e-06	5.87614854e-05	2.74e-07	5.87617812e-05	3.01e-07	5.87617621e-05	5.87617745e-05
20	1.01e-05	6.52293674e-05	2.92e-07	6.52297497e-05	3.38e-07	6.52297153e-05	6.52297363e-05

在实际的实验中, 带自适应策略的 ADMM 算法虽然能得到比 FPTA 算法更精确的解, 但迭代时间会比 FPTA 算法更长, 迭代次数也更多。

4. 结论

本文提出了一种带自适应策略的精确 ADMM 算法, 用于求解分式 l_1 正则化优化问题。该算法核心在于设计了自适应惩罚参数与正则化参数的动态调整机制, 从而较好地解决了标准 ADMM 方法中普遍存在的参数敏感问题。其中, 自适应 ρ 策略通过平衡原始残差与对偶残差的下降速度, 加快了算法的收敛速度。而自适应 λ 策略能平衡目标函数中数据拟合项与稀疏正则项之间的关系, 优化迭代过程。仿真实验结果显示, 在稀疏信号恢复任务中, 本算法表现出良好的性能。尤其在中小规模测试矩阵上, 其恢复成功率与 FPTA 算法相近, 但所求得的解在相对误差上更低, 更逼近真实的稀疏信号。

带自适应策略的精确 ADMM 算法虽然在求最优解方面表现优异, 但仍存在一些局限, 例如在迭代时间和迭代次数上会落后于 FPTA 算法。未来的研究可以通过完善该算法的收敛性分析, 探索有效的加速策略, 推广到其他的正则化问题等方面, 进一步提高算法的性能和实际应用价值。

基金项目

本研究受到了重庆市自然科学基金(项目编号: CSTB2022NSCQ-MSX0409)和重庆市教育委员会科技项目(基金编号: KJZD-M202201303)的资助。

参考文献

- [1] Beck, A. and Teboulle, M. (2009) A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. *SIAM Journal on Imaging Sciences*, **2**, 183-202. <https://doi.org/10.1137/080716542>
- [2] Wright, J., Yang, A. Y., Ganesh, A., Sastry, S. S. and Ma, Y. (2009) Robust Face Recognition via Sparse Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **31**, 210-227. <https://doi.org/10.1109/tpami.2008.79>
- [3] Ye, J. and Liu, J. (2012) Sparse Methods for Biomedical Data. *ACM SIGKDD Explorations Newsletter*, **14**, 4-15. <https://doi.org/10.1145/2408736.2408739>
- [4] Meinshausen, N. and Yu, B. (2009) Lasso-Type Recovery of Sparse Representations for High-Dimensional Data. *The Annals of Statistics*, **37**, 246-270. <https://doi.org/10.1214/07-aos582>
- [5] 崔安刚. 基于分式函数的范数优化的理论与 DC 算法研究[D]: [硕士学位论文]. 西安: 西安工程大学, 2016.
- [6] Daubechies, I., Defrise, M. and De Mol, C. (2004) An Iterative Thresholding Algorithm for Linear Inverse Problems

- with a Sparsity Constraint. *Communications on Pure and Applied Mathematics*, **57**, 1413-1457.
<https://doi.org/10.1002/cpa.20042>
- [7] Osborne, M.R. (1985) Finite Algorithms in Optimization and Data Analysis. Wiley.
- [8] Yazdanpanah, H., Carini, A. and Lima, M.V.S. (2019) L₀-Norm Adaptive Volterra Filters. 2019 27th European Signal Processing Conference (EUSIPCO), A Coruna, 2-6 September 2019, 1-5.
<https://doi.org/10.23919/eusipco.2019.8903013>
- [9] Li, H., Zhang, Q., Cui, A. and Peng, J. (2020) Minimization of Fraction Function Penalty in Compressed Sensing. *IEEE Transactions on Neural Networks and Learning Systems*, **31**, 1626-1637. <https://doi.org/10.1109/tnnls.2019.2921404>
- [10] Cui, A., He, H. and Yang, H. (2024) A Fast Algorithm for Sparse Signal Recovery via Fraction Function. *Electronics Letters*, **60**, e13243. <https://doi.org/10.1049/ell2.13243>
- [11] Wang, H., Kong, L. and Tao, J. (2017) The Linearized Alternating Direction Method of Multipliers for Sparse Group LAD Model. *Optimization Letters*, **13**, 505-525. <https://doi.org/10.1007/s11590-017-1180-3>
- [12] He, B.S., Yang, H. and Wang, S.L. (2000) Alternating Direction Method with Self-Adaptive Penalty Parameters for Monotone Variational Inequalities. *Journal of Optimization Theory and Applications*, **106**, 337-356.
<https://doi.org/10.1023/a:1004603514434>