Operations Research and Fuzziology 运筹与模糊学, 2025, 15(6), 100-108
Published Online December 2025 in Hans. https://www.hanspub.org/journal/orf
https://doi.org/10.12677/orf.2025.156260

Hans汉斯

对数和正则化的带保护条件的加速ADMM算法

曾子昕[1]，黄　华[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16日；录用日期：2025年12月10日；发布日期：2025年12月17日

摘　要

本文针对稀疏信号恢复问题，提出了一种基于对数和正则化模型的带有保护条件的Nesterov加速ADMM算法(nADMMgd)。该算法通过引入Nesterov加速技术有效提高了收敛速率，并利用保护条件保障了算法的稳定性。数值实验表明，nADMMgd算法在稀疏信号恢复问题上表现出色，能够在较短的时间内达到更优的函数值，且在计算大规模数据时具有良好的表现。实验验证了保护条件对计算效率方面提升的积极作用，还验证了nADMMgd对比目前用于计算对数和正则化的算法得到的结果更加精确。总体而言，本文提出的算法在稀疏信号恢复问题中具有显著的优势。

关键词

ADMM，对数和正则化，Nesterov，保护条件

Log-Sum Accelerated ADMM Algorithm with Protection Conditions

Zixin Zeng[1], Hua Huang[2*], Minghua Li[2]

[1]School of Mathematics and Statistics, Chongqing Jiaotong University, Chongqing
[2]School of Mathematics and Artificial Intelligence, Chongqing University of Arts and Sciences, Chongqing

Received: November 16, 2025; accepted: December 10, 2025; published: December 17, 2025

Abstract

This paper proposes a Nesterov-accelerated ADMM algorithm with protection conditions, referred to as nADMMgd, for sparse signal recovery based on logarithmic and regularization models. By incorporating Nesterov acceleration techniques, the algorithm achieves an improved convergence rate, while the integration of protective conditions enhances its numerical stability. Numerical ex-

*通讯作者。

文章引用: 曾子昕, 黄华, 李明华. 对数和正则化的带保护条件的加速ADMM算法[J]. 运筹与模糊学, 2025, 15(6): 100-108. DOI: 10.12677/orf.2025.156260

periments demonstrate that nADMMgd performs effectively in sparse signal recovery tasks, attaining superior objective function values within reduced computational time and exhibiting robust performance on large-scale datasets. The experimental results confirm the positive impact of the protective conditions on computational efficiency and further validate that nADMMgd yields more accurate solutions compared to state-of-the-art algorithms currently employed for logarithmic and regularized optimization. Overall, the proposed method presents significant advantages in addressing sparse signal recovery problems.

Keywords

ADMM, Log-Sum Regularization, Nesterov, Protection Conditions

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

在如今信息爆炸的时代，人们对信息的需求量暴增所面临的数据量与数据规模越来越大，运用传统的 Shannon-Nyquist (香农采样定理)处理数据的局限性越发明显[1]。在该背景下，压缩感知技术应运而生。Donoho [2]、Candès 和 Tao [3]基于信号的稀疏性和可压缩性提出的压缩感知技术(CS)，其在处理信号稀疏和可压缩数据时非常有用。在一些实际问题中，高维信号通过压缩处理通常可以表示为稀疏或者近似稀疏的向量，进而由少量的观测值恢复原始信号，即可以表示为如下线性方程组解 $b=Ax+e$，其中 $b\in\mathbb{R}^m$ 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，$e\in\mathbb{R}^m$ 为噪声向量，根据检测过程中是否含有噪声向量 e，该问题可以表示为下述 ℓ_0 范数极小化问题：

$$\min_{x\in\mathbb{R}^n}\left\{\|x\|_0 : Ax=b \text{ 或 } \|Ax-b\|_2\le\varepsilon\right\},$$

其中 $\varepsilon>0$ 是噪声水平，$\|x\|_0$ 表示 x 信号中非零元素的个数，通常我们称之为 ℓ_0 范数，它通常用来刻画向量的稀疏度。当解的噪声水平未知时，可以通过下述 ℓ_0 范数正则化问题来恢复稀疏信号：

$$\min_{x\in\mathbb{R}^n}\frac{1}{2}\|Ax-b\|_2^2+\lambda\|x\|_0,$$

其中 $\lambda>0$ 为正则化参数，该 ℓ_0 范数正则化问题是一个 NP-难的问题[4]，直接进行求解困难，为克服这一困难，一般对 ℓ_0 范数进行凸松弛，转化为 ℓ_1 范数正则化问题，ℓ_1 范数正则化问题也被称为 Least Absolute Shrinkage and Selection Operator (LASSO)模型[5]，得益于 ℓ_1 范数的凸性，求解 ℓ_1 范数松弛问题变得可行。不幸的是 ℓ_1 正则化问题得到的解稀疏性往往不够，特别是对于压缩感知，它会导致过度惩罚的情况。因此进一步的改进是必要的。近年来，研究者们提出了使用非凸松弛函数来近似 ℓ_0 范数，例如本文研究的对数和正则化模型[6]：

$$\min_x\left\{\frac{1}{2}\|Ax-b\|_2^2+\lambda\sum_{i=1}^n\log\left(\frac{|x_i|}{\epsilon}+1\right)\right\},\tag{1}$$

其中 $\lambda>0$ 为正则化参数，用于平衡系统数据拟合的准确性和解的稀疏性，x_i 是 x 中的第 i 项，ϵ 是确保函数定义良好的正参数。研究表明当 $\epsilon=0$ 时，对数和函数最小化是 ℓ_0 范数最小化的有效近似。由于其增强稀疏性的良好性质，对数和正则化得到了广泛应用。

Zhou [7]等人提出了一种基于对数和正则化的迭代阈值算法，该算法通过构造对数和函数的导数解析表达式，推导出相应的性质定理，进而提出了一种能够快速收敛到局部最小值的优化算法。Yu [8]等人受外推技术在加速一阶方法中的成功启发，研究了广泛使用的外推技术将其并入迭代重加权 L_1 算法。往往在高维问题或稀疏性较弱的情况下，以上算法达不到较精确的解。Zhang [9]提出了一种结合 L_1-L_2 惩罚函数的近端算子解析解和 v 加速技术的 ADMM 算法，得到的效果显著。Wang [10]等人建立 ADMM 在非凸目标函数和线性等式约束下的收敛理论，覆盖更广泛的非凸应用。Zeng [11]等人提出了一种名为 ASVRG-ADMM 的加速随机 ADMM 算法，结合外推加速(Nesterov 外推的改进版)和方差缩减技术，用于解决非凸非光滑的优化问题，且其证明了在 KL (Kurdyka-Łojasiewicz)性质下，序列几乎必然以线性速率收敛到临界点。因此，基于以上研究，本文通过研究 Buccini [12]等人提出的一种带保护条件的加速 ADMM 算法以及 Nesterov [13]提出的 Nesterov 加速梯度法，这里的 Nesterov 加速是一种用于一阶梯度下降优化的加速方法，提出了一种带保护条件的 Nesterov 加速 ADMM 算法。结合 Nesterov 外推的加速和保护条件的稳定性，以至于在精度上提升对数和正则化的稀疏优化效果。

2. 带保护条件的 Nesterov 加速 ADMM 算法

ADMM 是一种用于求解可分离优化问题的迭代算法。它特别适用于目标函数可以分解为两部分的和，且每部分只依赖于一个变量子集的情况。考虑如下问题：

$$\min_{x,z}\left\{f(x)+g(z)\right\}\ \text{s.t. } Ax+Bz=c,$$

下面给出 ADMM 的迭代格式，首先写出问题的增广拉格朗日函数：

$$L_\rho(x,z,u)=f(x)+g(z)+u^{\mathrm{T}}(Ax+Bz-c)+\frac{\rho}{2}\|Ax+Bz-c\|_2^2,$$

其中 $\rho>0$ 是二次罚项的系数。常见的求解带约束问题的增广拉格朗日函数法为如下更新：

$$(x^{k+1},z^{k+1})=\arg\min_{x,z}L_\rho(x,z,u^k),$$

$$u^{k+1}=u^k+\rho\left(Ax^{k+1}+Bz^{k+1}-c\right),$$

在实际的求解问题过程中，第一步迭代同时对 x 和 z 进行更新比较困难，但固定其中一个变量求解关于另外一个变量的极小化问题比较简单，因此我们考虑对 x 和 z 交替求极小，而这就是交替方向乘子法的思路，以上的求解思路可以总结为如下：

$$x^{k+1}=\arg\min_{x}L_\rho(x,z^k,u^k),$$

$$z^{k+1}=\arg\min_{z}L_\rho(x^{k+1},z\ ,u^k),$$

$$u^{k+1}=u^k+\rho\left(Ax^{k+1}+Bz^{k+1}-c\right).$$

在本文研究的对数和正则化模型中，我们先将问题(1)重新表述为：

$$\min_{x,z}\frac{1}{2}\|Ax-b\|_2^2+\lambda\sum_{i=1}^{n}\left(\left|\frac{z_i}{\epsilon}+1\right|\right)\quad\text{s.t. } x-z=0, \tag{2}$$

增广拉格朗日函数为：

$$L_\rho(x,z,u)=\frac{1}{2}\|Ax-b\|_2^2+\lambda\sum_{i=1}^{n}\log\left(\frac{|z_i|}{\epsilon}+1\right)+u^{\mathrm{T}}(x-z)+\frac{\rho}{2}\|x-z\|_2^2,$$

ADMM 的迭代公式为：

$$x^{k+1}=\arg\min_{x}\left\{\frac{1}{2}\|Ax-b\|_2^2+\left(u^k\right)^{\mathrm{T}}\left(x-z^k\right)+\frac{\rho}{2}\left\|x-z^k\right\|_2^2\right\},$$

$$z^{k+l}\in\operatorname{argmin}_{z}\left\{\lambda\sum_{i=l}^{n}\log\left(\frac{|z_i|}{\epsilon}+1\right)+\left(u^k\right)^{\mathrm{T}}\left(x^{k+l}-z\right)+\frac{\rho}{2}\left\|x^{k+l}-z\right\|_2^2\right\},$$

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

ADMM 的迭代步骤为：

步骤 1： x 更新(固定 z^k 和 u^k)

$$x^{k+1}=\arg\min_{x}\left\{\frac{1}{2}\|Ax-b\|_2^2+\left(u^k\right)^{\mathrm{T}}\left(x-z^k\right)+\frac{\rho}{2}\left\|x-z^k\right\|_2^2\right\},$$

简化后：

$$x^{k+1}=\arg\min_{x}\left\{\frac{1}{2}\|Ax-b\|_2^2+\frac{\rho}{2}\left\|x-\left(z^k-\frac{u^k}{\rho}\right)\right\|_2^2\right\},$$

该问题的解由以下线性方程给出：

$$x^{k+1}=\left(A^{\mathrm{T}}A+\rho I\right)^{-1}\left(A^{\mathrm{T}}b+\rho\left(z^k-\frac{u^k}{\rho}\right)\right),$$

步骤 2： z 更新(固定 x^{k+1} 和 u^k)

$$z^{k+1}\in\arg\min_{z}\left\{\lambda\sum_{i=1}^{n}\log\left(\frac{|z_i|}{\epsilon}+1\right)-\left(u^k\right)^{\mathrm{T}}z+\frac{\rho}{2}\left\|x^{k+1}-z\right\|_2^2\right\},$$

这里目标函数可分解为 n 个独立标量的函数，可逐项求解：

$$z_i^{k+1}\in\arg\min_{z_i}\left\{\lambda\log\left(\frac{|z_i|}{\epsilon}+1\right)+\frac{\rho}{2}\left(z_i-w_i^k\right)^2\right\},$$

其中，

$$w^k=x^{k+1}+\frac{u^k}{\rho},$$

为了求解这个独立问题，定义函数 $h\left(z_i\right)$：

$$h\left(z_i\right)=\lambda\log\left(\frac{|z_i|}{\epsilon}+1\right)+\frac{\rho}{2}\left(z_i-w_i^k\right)^2,$$

经过对 $h\left(z_i\right)$ 求导，我们得到：

$$h'\left(z_i\right)=\frac{\lambda\cdot\operatorname{sign}\left(z_i\right)}{|z_i|+\epsilon}+\rho\left(z_i-w_i^k\right),$$

根据一阶最优性条件令导数为零，我们得到变量 z_i^{k+1} 的解析解为：

$$z_i^{k+1}=\begin{cases}0 & \left|w_i^k\right|\le\dfrac{\lambda}{\rho\epsilon}\\ \dfrac{\left(w_i^k-\epsilon\right)+\sqrt{\left(w_i^k+\epsilon\right)^2-4\dfrac{\lambda}{\rho}}}{2} & w_i^k>\dfrac{\lambda}{\rho\epsilon}\text{且 }D\ge 0\\ \dfrac{\left(w_i^k+\epsilon\right)-\sqrt{\left(w_i^k-\epsilon\right)^2-4\dfrac{\lambda}{\rho}}}{2} & w_i^k<-\dfrac{\lambda}{\rho\epsilon}\text{且 }D\ge 0\\ 0 & \text{其他情况,}\end{cases}$$

其中 $D=\left(w_i^k+\epsilon\right)^2-\dfrac{4\lambda}{\rho}$ 确保判别式非负。

步骤 3：乘子更新(固定 x^{k+1} 和 z^{k+1})

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

基于上述的迭代步骤，我们通过引入外推技巧和保护条件，进一步加快算法的计算效率和提升算法的稳定性。在外推技巧方面，我们采用了 Nesterov 加速的思想，在每一步迭代中，我们不仅计算当前原始更新 $(\overline{z},\overline{u})$，还利用一个外推路径 $(\hat{z},\hat{u})$ 来更新 x，外推参数 θ 由加速参数 α 更新，具体表现为：

$$\alpha_k=\frac{1+\sqrt{1+4\alpha_{k-1}}}{2},\theta_k=\frac{\alpha_{k-1}-1}{\alpha_k},$$

其本质是迭代阈值算法(ISTA)的快速版本。

在加速 ADMM 算法中，Nesterov 外推可以显著提高收敛速度，尤其是在早期迭代中。然而，在某些情况下，外推可能导致算法不稳定甚至发散，保护条件的作用是在外推可能导致不稳定的情况下，自动回退到原始 ADMM 更新。该机制具体表现在当条件 $\gamma_{k+1}<\gamma_0\eta^k$ 被满足时，采用外推。否则，我们保留未经过 θ 加速计算的原始更新值。这样，当外推导致残差度量 γ 过大(超过递减阈值)时，我们就放弃外推，使用原始的 ADMM 更新，并且重置加速参数，重新开始加速过程。加速机制的目的是加快收敛速度，而保护条件则确保算法在发散风险时回退到原始更新，从而保证稳定性。

首先我们定义混合残差：

$$\gamma_{k+1}=\delta^{-1}\left\|\hat{u}_{k+1}-u_k\right\|_2^2+\delta\left\|\hat{z}_{k+1}-z_k\right\|_2^2,$$

其中 z^k，u^k 为上一步的迭代点，其次定义一个递减阈值 η，随着迭代次数 k 的增加，η^k 指数衰减，因此 $\gamma_0\eta^k$ 也会逐渐减小，这意味着在早期的迭代中，我们允许较大的外推步，而随着迭代进行，我们对外推的要求越来越严格。在标准的 ADMM 中，我们定义残差有原始残差 $r^k=x^k-z^k$ 和对偶残差 $s^k=\rho\left(z^k-z^{k-1}\right)$，在此框架下，对偶变量 u 的更新与原始残差有关，而变量 z 的更新与对偶残差有关，因此 γ_{k+1} 同时捕捉了原始变量和对偶变量的变化。其中 η 在这一过程中起关键性作用，其值选取通常在 0.8 到 0.99 之间。通过这种方式，算法在大多数情况下可以享受 Nesterov 加速带来的快速收敛，同时避免发散风险。

相对于非精确加速算法，保护条件的主要思想是通过监控外推步的适用性来确保稳定性。如果外推步可能导致的不稳定(即不满足保护条件)，则拒绝外推步，回退到标准的非加速迭代。而非精确加速的主要思想是在迭代过程中，允许子问题(如 ADMM 算法中的 z-更新)的求解存在一定的误差(即不要求精确求解)，并通过控制误差的衰减速度来确保整体的收敛性，然而非精确加速需要设计合适的误差控制策略，

如果误差控制的不好，可能会导致收敛变得更慢甚至发散，其理论分析更复杂。因此对于对数和正则化该类非凸问题，保护条件的设立更能保证全局的收敛性。

基于以上内容，我们给出 nADMMgd 算法。

算法 1. 带保护条件的 n 加速 ADMM 算法(nADMMgd)

初始化：$x_0 = y_0 = u_0 = 0$，$\bar{y}_0 = \bar{u}_0 = 0$，$k = 0$。

步骤 1：

$$x^{k+1} = \arg\min_x \left\{ \frac{1}{2}\|Ax-b\|_2^2 + \left(u^k\right)^{\mathrm{T}}\left(x - z^k\right) + \frac{\rho}{2}\left\|x - z^k\right\|_2^2 \right\}$$

$$\bar{z}^{k+1} \in \arg\min_z \left\{ \lambda \sum_{i=1}^{n} \log\left(\frac{|z_i|}{\epsilon} + 1 \right) - \left(u^k\right)^{\mathrm{T}} z + \frac{\rho}{2}\left\|x^{k+1} - z\right\|_2^2 \right\}$$

$$\bar{u}^{k+1} = u^k + \rho\left(x^{k+1} - z^{k+1}\right)$$

$$\hat{z}^k = \bar{z}^k + \theta^k\left(\bar{z}^k - \bar{z}^{k-1}\right)$$

$$\hat{u}^k = \bar{u}^k + \theta^k\left(\bar{u}^k - \bar{u}^{k-1}\right)$$

$$\gamma_{k+1} = \delta^{-1}\left\|\hat{u}_{k+1} - u_k\right\|_2^2 + \delta\left\|\hat{z}_{k+1} - z_k\right\|_2^2$$

步骤 2：

If $\gamma^{k+1} < \gamma^0 \eta^k$

$z^{k+1} = \hat{z}^{k+1}$

$u^{k+1} = \hat{u}^{k+1}$

Else

$z^{k+1} = \bar{z}^{k+1}$

$u^{k+1} = \bar{u}^{k+1}$

end

步骤 3：如果不满足终止条件则 $k = k+1$ 转至步骤 1

3. 数值实验

在本节中，我们进行数值实验来研究 nADMMgd 算法在解决对数和正则化最小二乘问题的效果，所有的代码都使用 Matlab 编写，并且在具有 12th Gen Intel(R) Core(TM) i5-12490F 3.00 GHz 和 16GB RAM 的 64 位的 PC 上在 Matlab 2024a 中进行实验。其中，对数和正则化最小二乘问题为：

$$\min_x \left\{ \frac{1}{2}\|Ax-b\|_2^2 + \lambda \sum_{i=1}^{n} \log\left(\frac{|x_i|}{\epsilon} + 1 \right) \right\}.$$

在数值实验中，我们设置正则化参数为 $\lambda = 5\mathrm{e}-4$、$\epsilon = 0.1$ 来比较 nADMMgd 算法与现有算法的性能。针对问题，首先我们设置了一个具有独立同分布的标准高斯矩阵 A，其大小为 $720i \times 2560i$，然后将该矩阵归一化，使其每一列的范数均为单位范数。并生成一个 $2560j$ 维的稀疏向量 y，该向量包含 $80i$ 个随机位置、数值的非零元素，且这些非零元素服从标准正态分布，其中 $i = 1,2,3,\cdots,10$ 表示实验的不同规模。观测向量 b 通过 $b = Ay + 0.01\cdot\omega$ 生成，其中 $\omega \in \mathbb{R}^m$ 是一个独立同分布的标准高斯噪声向量。我们使用 10 种不同规模(即不同 i)的矩阵 A 进行实验。对于每种规模，独立重复实验 20 次，并在每次实验中记录 CPU

时间和算法停止迭代时的目标函数值作为性能评估指标。

对于 nADMMgd 算法，经验调优后确定外推参数 $\alpha=1$ 和保护阈值参数 $\eta=0.95$。此外，算法的终止条件设置为：

$$\|x_k-z_k\|_2<10^{-5} \text{ 或 } \|z_k-z_{k-1}\|_2<10^{-5}.$$

在表 1 中我们设置了不同的保护阈值参数 η，实验表明当 $\eta=0.95$ 时的 nADMMgd 算法收敛最快、迭代次数最少，运行时间最短，且在两种不同规模下均适用，因此该参数具有良好的鲁棒性。在算法中参数 δ 的取值通常与罚参数 ρ 相关，因此不再做参数 δ 的敏感性分析。

Table 1. The influence of parameter η in nADMMgd algorithm on the algorithm
表 1. nADMMgd 算法中 η 参数对算法的影响

规模	η	Iters	Time(s)	Fval
720*2560	0.80	1636	1.7	9.33028e−02
	0.85	2072	2.3	9.76328e−02
	0.90	1814	1.9	9.76327e−02
	0.95	**411**	**0.4**	**9.76253e−02**
	0.99	653	0.7	9.76316e−02
3600*12800	0.80	1876	84.1	4.65579e−01
	0.85	1612	74.1	4.62300e−01
	0.90	1397	63.8	4.62300e−01
	0.95	**446**	**21.6**	**4.65578e−01**
	0.99	889	41.2	4.62299e−01

Table 2. Comparison of accelerated ADMM algorithms under protected and unprotected conditions
表 2. 有保护与无保护条件下加速 ADMM 算法比较

Method	M	Iter	Time(s)	Fval
nADMMgd	720*2560	382	**0.41**	9.330253e−02
nADMM		679	0.71	9.330208e−02
nADMMgd	1440*5120	424	**3.03**	1.866190e−01
nADMM		780	5.55	1.866189e−01
nADMMgd	2160*7680	440	**7.24**	2.793258e−01
nADMM		842	13.64	2.793258e−01
nADMMgd	2880*10240	447	**13.49**	3.738720e−01
nADMM		873	25.74	3.738719e−01
nADMMgd	3600*12800	454	**22.34**	4.655781e−01
nADMM		909	43.46	4.655781e−01

这里我们通过实验展示了带保护条件的加速 ADMM 算法与无保护的加速 ADMM 算法的效果，通过表 2 我们可以看到带保护的算法相对于无保护的算法收敛速率更快、迭代次数更少，同时保持较高的恢复精度。

Table 3. Noise signal recovery result
表 3. 噪声信号恢复结果

Problem Size		Time(s)				Fval			
m	**n**	IRL_1e_1	IRL_1e_2	IRL_1e_3	**nADMMgd**	IRL_1e_1	IRL_1e_2	IRL_1e_3	**nADMMgd**
720	2560	0.05	0.07	0.05	0.4	9.330500e−02	9.330224e−02	9.330273e−02	**9.330253e−02**
1440	5120	0.56	0.73	0.60	2.95	1.866253e−01	1.866193e−01	1.866202e−01	**1.866190e−01**
2160	7680	1.48	1.96	1.62	7.10	2.793357e−01	2.793262e−01	2.793278e−01	**2.793258e−01**
2880	10,240	2.87	3.80	3.31	13.20	3.738860e−01	3.738724e−01	3.738746e−01	**3.738720e−01**
3600	12,800	4.80	6.35	5.17	21.69	4.655960e−01	4.655788e−01	4.655817e−01	**4.655781e−01**
4320	15,460	6.77	8.94	7.38	30.68	5.634407e−01	5.634210e−01	5.634245e−01	**5.634202e−01**
5040	17,920	9.07	12.08	9.88	41.10	6.531319e−01	6.531102e−01	6.531141e−01	**6.531094e−01**
5760	20,480	11.83	15.70	12.87	55.65	7.495698e−01	7.495439e−01	7.495483e−01	**7.495428e−01**
6480	23,040	14.97	19.84	16.27	71.27	8.398036e−01	8.397736e−01	8.397785e−01	**8.397724e−01**
7200	25,600	18.31	24.35	19.98	88.06	9.346740e−01	9.346408e−01	9.346464e−01	**9.346393e−01**

根据我们的实验结果，表 3 显示了各算法在 $\lambda = 5\mathrm{e}-4$， $\epsilon = 0.1$ 参数选取下的迭代时间与收敛时的目标函数值，其中加黑字体为最优的结果。从结果中可以看出，随着问题规模增大，nADMMgd 在收敛时的目标函数值上的优势保持稳定，且与第二优的算法的差距基本保持在大约 1e−5 到 1e−4 量级。因此，nADMMgd 算法在优化精度上具有明显优势，尽管计算时间较长。因此，在需要高精度的场景下，可以选择 nADMMgd 算法。

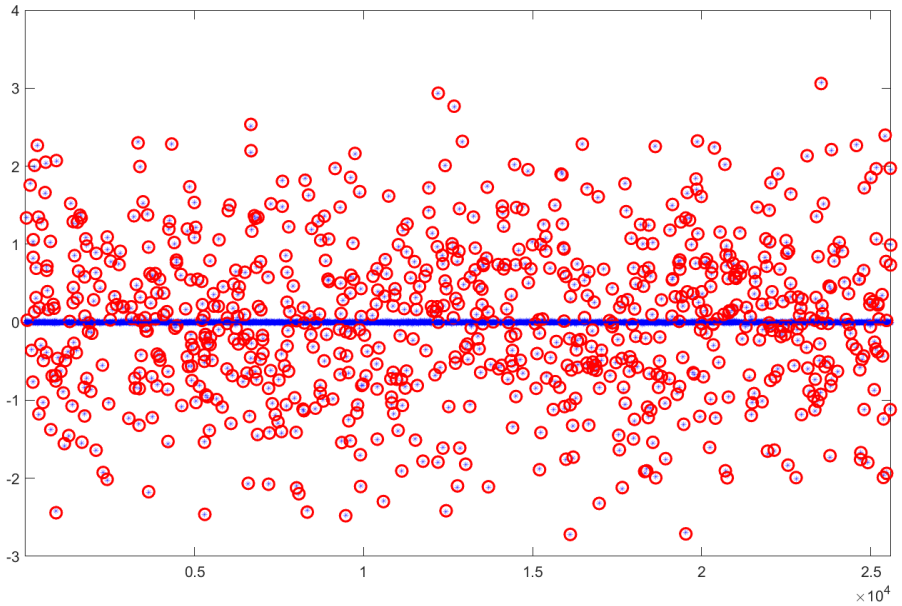

Figure 1. nADMMgd recovery results
图 1. nADMMgd 恢复结果

最后为了证明通过我们的算法恢复初始稀疏解的能力，我们绘制了 nADMMgd 求解(2)的获得的真实信号与恢复信号，图 1 表明我们的算法得到的恢复信号(用蓝色星号标记)接近真实信号(用红色圆圈标记)。

4. 结论

本文提出了一种带保护条件的 Nesterov 加速 ADMM 算法，用于求解对数和正则化模型。通过数值实验验证了该算法在稀疏信号恢复上恢复精度的优势。研究表明 nADMMgd 算法在不同规模的测试矩阵上均表现出优异的性能，尤其是在大规模问题上，其计算精度明显优于 IRL1 类算法。尽管 nADMMgd 算法在精度方面出色，但仍存在一些不足，例如其在计算效率上远低于普通算法。由于对数和正则化是非凸函数，其目标函数可能存在多个局部极小值，导致算法收敛到不同的解，因此得到的解可能和目标函数值有偏差。对于非凸问题，传统的 ADMM 收敛性理论(针对凸问题)不再适用。虽然近年有工作研究非凸 ADMM 的收敛性，但通常需要较强的假设，且收敛速度难以保证。未来将在算法的收敛性分析方面进行研究，如是否能在 KL 条件的假设下证明其收敛性，为算法的理论层面提供保障。

基金项目

本研究受到了重庆市自然科学基金(项目编号：CSTB2024NSCQ-LZX0091)和重庆市教育委员会科技项目(项目编号：KJZD-M202201303)的资助。

参考文献

[1] 胡耀华, 李昱帆, 刘艳艳, 等. 结构稀疏优化模型的理论与算法[J]. 中国科学: 数学, 2024, 54(7): 1045-1070.

[2] Donoho, D.L. (2006) Compressed Sensing. *IEEE Transactions on Information Theory*, **52**, 1289-1306. https://doi.org/10.1109/tit.2006.871582

[3] Candes, E.J. and Tao, T. (2005) Decoding by Linear Programming. *IEEE Transactions on Information Theory*, **51**, 4203-4215. https://doi.org/10.1109/tit.2005.858979

[4] Natarajan, B.K. (1995) Sparse Approximate Solutions to Linear Systems. *SIAM Journal on Computing*, **24**, 227-234. https://doi.org/10.1137/s0097539792240406

[5] Tibshirani, R. (1996) Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society Series B*: *Statistical Methodology*, **58**, 267-288. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x

[6] Candès, E.J., Wakin, M.B. and Boyd, S.P. (2008) Enhancing Sparsity by Reweighted L1 Minimization. *Journal of Fourier Analysis and Applications*, **14**, 877-905. https://doi.org/10.1007/s00041-008-9045-x

[7] Zhou, X., Liu, X., Zhang, G., Jia, L., Wang, X. and Zhao, Z. (2023) An Iterative Threshold Algorithm of Log-Sum Regularization for Sparse Problem. *IEEE Transactions on* Circuits *and Systems for Video Technology*, **33**, 4728-4740. https://doi.org/10.1109/tcsvt.2023.3247944

[8] Yu, P. and Pong, T.K. (2019) Iteratively Reweighted L1 Algorithms with Extrapolation. *Computational Optimization and Applications*, **73**, 353-386. https://doi.org/10.1007/s10589-019-00081-1

[9] 张海林. 带有保护条件的 v 加速 ADMM 算法[J]. 理论数学, 2025, 15(4): 298-308.

[10] Wang, Y., Yin, W. and Zeng, J. (2019) Global Convergence of ADMM in Nonconvex Nonsmooth Optimization. *Journal of Scientific Computing*, **78**, 29-63. https://doi.org/10.1007/s10915-018-0757-z

[11] Zeng, Y., Wang, Z., Bai, J. and Shen, X. (2024) An Accelerated Stochastic ADMM for Nonconvex and Nonsmooth Finite-Sum Optimization. *Automatica*, **163**, Article 111554. https://doi.org/10.1016/j.automatica.2024.111554

[12] Buccini, A., Dell'Acqua, P. and Donatelli, M. (2020) A General Framework for ADMM Acceleration. *Numerical Algorithms*, **85**, 829-848. https://doi.org/10.1007/s11075-019-00839-y

[13] Nesterov, Y. (2013) Gradient Methods for Minimizing Composite Functions. *Mathematical Programming*, **140**, 125-161. https://doi.org/10.1007/s10107-012-0629-5