



论 文

基于 $S_{1/2}$ 建模的稳健稀疏-低秩矩阵分解饶过^{①②}, 彭毅^{①②}, 徐宗本^{①②*}

① 西安交通大学信息与系统科学研究所, 西安 710049

② 西安交通大学智能网络与网络安全教育部重点实验室, 西安 710049

* 通信作者. E-mail: zbxu@mail.xjtu.edu.cn

收稿日期: 2012-09-02; 接受日期: 2012-10-18

国家自然科学基金(批准号: 61075054, 60975036, 11171272)和国家重点基础研究发展计划(973计划)(批准号: 2013CB329404)资助项目

摘要 为实现稳健的稀疏-低秩矩阵分解, 本文首次引入矩阵的 $S_{1/2}$ 范数以诱导矩阵的低秩性来构建新模型, 并在 ADMM 算法框架下设计了高效的交替阈值迭代算法. 该算法采用增广 Lagrange 乘子技术, 在迭代过程中交替更新低秩矩阵和稀疏矩阵. 由于这两个矩阵的最优更新具有显式形式, 算法整体的计算精度和时间代价得以控制. 大量的数值模拟实验说明: 相较于目前最好的不精确 ALM 算法, 交替阈值迭代算法的迭代次数与时间代价大幅降低, 对噪声更为稳健, 分解出的低秩矩阵的秩与稀疏矩阵的稀疏度更接近于真实值. 在对监控视频进行背景建模这一实际问题中, 交替阈值迭代算法得到的背景矩阵更为低秩, 更符合问题先验, 且时间代价相较于不精确 ALM 算法降幅高达一个数量级, 这说明新模型与算法能有效解决相关实际问题.

关键词 $S_{1/2}$ 范数 稀疏-低秩矩阵分解 快速稳健 交替阈值迭代

1 引言

在工程应用中, 许多复杂的系统通常是由多个简单的系统组成. 为更好地理解复杂系统的行为与性质, 人们往往利用复杂系统的可分解性将其分解成若干简单系统来进行研究. 特别地, 当复杂系统的矩阵表示是一个稀疏矩阵与一个低秩矩阵之和时, 分解该复杂系统可描述成稀疏-低秩矩阵分解问题: 已知矩阵 D 可表示成低秩矩阵 A 与稀疏矩阵 E 之和, A 的秩信息与 E 的稀疏结构信息未知, 如何由 D 准确地分解出矩阵 A 与 E ? 该问题的数学模型如下:

$$\min_{A, E \in \mathbb{R}^{m \times n}} \text{rank}(A) + \lambda \|E\|_{l_0} \quad \text{s.t.} \quad D = A + E, \quad (1)$$

其中 $\text{rank}(A)$ 表示矩阵 A 的秩, $\|E\|_{l_0}$ 表示矩阵 E 的非零元个数即 l_0 范数, 参数 λ 则用于平衡矩阵 A 的低秩程度和矩阵 E 的稀疏程度.

显然, 稀疏-低秩矩阵分解问题通常是病态的 (NP-hard). 受压缩感知与统计领域相关研究工作^[1,2]的启发, 文献[3]基于“秩-稀疏不相关性”(rank-sparsity incoherence)给出了问题可解的条

引用格式: 饶过, 彭毅, 徐宗本. 基于 $S_{1/2}$ 建模的稳健稀疏-低秩矩阵分解. 中国科学: 信息科学, 2013, 43: 733–748, doi: 10.1360/112012-538

件, 文献 [4~6] 则提出通过求解如下凸优化问题实现主成分追踪 (principle component pursuit, PCP), 以准确分解出稀疏与低秩矩阵:

$$\min_{A, E \in \mathbb{R}^{m \times n}} \|A\|_* + \lambda \|E\|_{l_1} \quad \text{s.t.} \quad D = A + E, \quad (2)$$

这里 $\|\cdot\|_*$ 表示矩阵奇异值之和, 称为矩阵的核范数, $\|\cdot\|_{l_1}$ 表示矩阵所有元素绝对值之和.

然而, 实际中的系统往往会受噪声影响, 矩阵 D 不能严格地表示成低秩矩阵与稀疏矩阵之和. 为此, 文献 [7] 在假设 $\|D - A - E\|_F \leq \delta$ ($\delta > 0$) 的情况下提出稳定主成分追踪 (stable principle component pursuit, SPCP), 并证明求解优化问题 (3) 所得到的矩阵 A 与 E 的误差平方和以 $O(\delta^2)$ 为上界.

$$\min_{A, E \in \mathbb{R}^{m \times n}} \|A\|_* + \lambda \|E\|_{l_1} \quad \text{s.t.} \quad \|D - A - E\|_F \leq \delta. \quad (3)$$

为求解优化问题 (2) 与 (3), 文献 [7] 基于目标函数的一阶信息提出了阈值迭代框架, 但算法的收敛速度很慢, 且每次迭代都要进行一次奇异值分解, 无法快速求解大型实际问题. 此后, 文献 [8] 提出了直接求解原始问题的 APG 算法和求解相应偶问题的梯度上升算法, 并通过实验说明这两种算法在处理 1000×1000 的矩阵时相较于阈值迭代算法加速了 50 倍. 进一步地, 文献 [9] 基于 Lagrange 乘子技术提出了不精确 ALM 算法 (简称为 ALM 算法), 在显著提高计算精度的同时其计算速度是 APG 算法的 5 倍, 这是目前已知的求解大规模问题的最好算法.

稀疏-低秩矩阵分解问题可理解成从稀疏矩阵与低秩矩阵组成的过完备字典中寻找矩阵 D 的 (近似) 最简单表示 $A + E$, 其中稀疏矩阵 E 的简单程度用其非零元素的个数即矩阵的 l_0 范数来刻画, 可视为矩阵在一维意义下的稀疏度, 低秩矩阵 A 的简单程度则用矩阵秩即非零奇异值的个数来刻画, 可视为矩阵在二维意义下的稀疏性. PCP 与 SPCP 受到广泛关注的原因就在于: 用矩阵的 l_1 范数代替 l_0 范数来刻画矩阵的一维稀疏度, 用核范数即奇异值向量的 l_1 范数代替矩阵秩来刻画矩阵的二维稀疏度, 从而得到原始组合优化问题 (1) 的最紧凸松弛形式.

近年来已发展成熟的 $l_{1/2}$ 正则化理论^[10,11] 则说明: 相较于常用的 l_1 范数, $l_{1/2}$ 范数可进一步诱导向量的稀疏性, 在稀疏信号重建中对噪声具有更强的稳健性. 为得到矩阵 D 更为简单的 (近似) 表示, 本文将向量的 $l_{1/2}$ 范数推广到矩阵, 分别用 $S_{1/2}$ 范数和 $l_{1/2}$ 范数来刻画矩阵二维与一维稀疏度, 提出如下实现 (近似) 稀疏-低秩矩阵分解的 $S_{1/2}$ 模型:

$$\min_{A, E \in \mathbb{R}^{m \times n}} \|A\|_{S_{1/2}}^{1/2} + \lambda \|E\|_{l_a}^a \quad \text{s.t.} \quad \|D - A - E\|_F \leq \delta, \quad (4)$$

其中 $\|A\|_{S_{1/2}} = (\sum_{i=1}^r \sigma_i^{1/2})^2$ 表示矩阵 A 所有奇异值 $\{\sigma_i\}_{i=1}^r$ 构成向量的 $l_{1/2}$ 范数, 对应于 Schatten _{p} 范数 $p = \frac{1}{2}$ 的情形, 本文简记为 $S_{1/2}$ 范数. $\|E\|_{l_a}^a = (\sum_{i=1}^m \sum_{j=1}^n |E_{ij}|^a)^{1/a}$ 表示矩阵 E 拉直向量的 l_a 范数, 参数 a 的选取可采用如下启发式的方法: $\delta > 0$ 表示在有噪声情况下对矩阵 D 进行近似分解, 取 $a = 1$ 即用 l_1 范数刻画矩阵的一维稀疏性; $\delta = 0$ 表示对矩阵 D 进行准确分解, 取 $a = \frac{1}{2}$ 即用 $l_{1/2}$ 范数进一步诱导向量的一维稀疏性.

考虑到 $S_{1/2}$ 模型的目标函数与约束条件关于稀疏矩阵与低秩矩阵是可分的, 本文基于 ADMM^[12] 算法思想提出交替阈值迭代算法. 该算法利用增广 Lagrange 乘子技术, 在迭代过程中采用交替投影的思想逐个更新低秩矩阵、稀疏矩阵和 Lagrange 乘子. 低秩矩阵与稀疏矩阵的更新需要求解两个非线性非凸优化问题, 本文通过将作用于向量的 Soft^[13] 与 Half^[10] 阈值算子推广到矩阵情形以给出其最优解的显式形式, 这从很大程度上保证了所设计算法的高精度与低时间代价特性.

与目前最好的 ALM 算法相比, 大量的模拟实验说明本文所提出的交替阈值迭代算法具有以下优点: 达到收敛所需迭代次数与时间代价大幅降低, 对噪声有更强的稳健性, 分解出的低秩矩阵的秩更接近于真实值, 同时算法的可靠性对矩阵 A 的低秩程度依赖更少. 另外, 在监控视频背景建模^[4]这一实际应用中, 交替阈值迭代算法能得到更为低秩的背景矩阵, 这一现象更符合建模时利用的先验, 且时间代价相较于 ALM 算法降幅高达一个数量级, 这对实际中海量视频数据的快速处理具有重要意义.

2 基于 ADMM 的交替阈值迭代算法

2.1 算法思想与迭代框架

ADMM 是一类用于求解分布式凸优化问题的简单有效算法, 它采用分解-协同过程 (decomposition-coordination procedure) 处理问题, 同时具备对偶上升算法的可分解性和增广 Lagrange 乘子法的收敛性, 一般用于求解如下形式的优化问题:

$$\min_{x,z} f(x) + g(z) \quad \text{s.t.} \quad Mx + Nz = c, \quad (5)$$

其中 $x \in \mathbb{R}^n$, $z \in \mathbb{R}^m$, $M \in \mathbb{R}^{p \times n}$, $N \in \mathbb{R}^{p \times m}$, f, g 为凸函数.

考虑问题的增广 Lagrange 乘子形式:

$$L_{\mu}(x, z, y) = f(x) + g(z) + y^{\top}(Mx + Nz - c) + \frac{1}{2}\mu\|Mx + Nz - c\|_2^2, \quad (6)$$

其中 y 表示线性等式约束的乘子, $\mu > 0$ 表示对不满足线性等式约束的惩罚因子, 也称为增广 Lagrange 参数, 则 ADMM 主要由以下 3 个子迭代过程构成:

$$\begin{cases} x^{k+1} = \arg \min_x L_{\mu}(x, z^k, y^k), \\ z^{k+1} = \arg \min_z L_{\mu}(x^{k+1}, z, y^k), \\ y^{k+1} = y^k + \mu(Mx^{k+1} + Nz^{k+1} - c). \end{cases} \quad (7)$$

显然, ADMM 与对偶上升法和乘子法非常相似, 由 x 极小化、 z 极小化和对偶变量 y 更新组成, 且对偶变量的更新步长与乘子法一样取为 μ .

虽然 ADMM 的收敛性理论建立在目标函数 $f(\cdot)$ 和 $g(\cdot)$ 为凸函数的基础上, 但其思想可推广到非凸优化问题的求解, 本文正是基于这样的推广来设计求解新模型 (4) 的交替阈值迭代算法. 我们先考虑新模型的增广 Lagrange 乘子形式:

$$L_{\mu}(A, E, Y) = \|A\|_{S_{1/2}}^{1/2} + \lambda\|E\|_{l_a}^a + \langle Y, D - A - E \rangle + \frac{1}{2}\mu\|D - A - E\|_F^2, \quad (8)$$

其中 $Y \in \mathbb{R}^{m \times n}$ 是线性约束乘子, $\mu > 0$ 表示对不满足等式约束 $D = A + E$ 的惩罚因子, 其取值依赖于模型 (4) 中正参数 δ , 特别地, 当 $\delta = 0$ 时 $\mu = \infty$. 为极小化 $L_{\mu}(A, E, Y)$, 我们通过迭代求解如下 3 个子问题:

$$\begin{cases} A_{k+1} = \arg \min_A L_{\mu}(A, E_k, Y_k), \\ E_{k+1} = \arg \min_E L_{\mu}(A_{k+1}, E, Y_k), \\ Y_{k+1} = Y_k + \mu(D - A_{k+1} - E_{k+1}). \end{cases} \quad (9)$$

通过对更新矩阵 A 和 E 所需求解子问题的目标函数进行分解与重组, 可得到如式 (9) 的简化形式:

$$\begin{cases} A_{k+1} = \arg \min_A \frac{1}{2} \mu \|A - (D - E_k + \frac{1}{\mu} Y_k)\|_F^2 + \|A\|_{S_{1/2}}^{1/2}, \\ E_{k+1} = \arg \min_E \frac{\mu}{2\lambda} \|E - (D - A_{k+1} + \frac{1}{\mu} Y_k)\|_F^2 + \|E\|_{l_a}^a, \\ Y_{k+1} = Y_k + \mu(D - A_{k+1} - E_{k+1}). \end{cases} \quad (10)$$

上述交替阈值迭代算法的关键是在每次迭代中交替更新低秩矩阵、稀疏矩阵和 Lagrange 乘子, 直到满足预先设定的收敛条件. 因此, 算法整体的计算精度与时间代价在很大程度上取决于式 (10) 中两个子问题的求解.

2.2 两个子问题的显式最优解

式 (10) 中更新稀疏矩阵和低秩矩阵所需求解的两个子问题可分别简化成如下形式:

$$\min_X \|X - W\|_F^2 + \lambda \|X\|_{l_a}^a, \quad (11)$$

$$\min_X \|X - W\|_F^2 + \lambda \|X\|_{S_{1/2}}^{1/2}. \quad (12)$$

问题 (11) 的求解比较简单, 文献 [9] 已证明 $a = 1$ 时的最优解为 $ST_{\lambda/2}(W)$, 即以 $\frac{\lambda}{2}$ 为阈值对矩阵 W 的每个元素进行 Soft 阈值收缩, 相应的 Soft 阈值算子 [12] 定义为

$$ST_{\lambda}(x) = \begin{cases} x - \text{sgn}(x)\lambda, & |x| > \lambda, \\ 0, & \text{其他}, \end{cases} \quad (13)$$

其中 $\text{sgn}(\cdot)$ 是符号函数.

由于在 $a = 1$ 与 $\frac{1}{2}$ 这两种情形下, 问题 (11) 的目标函数与约束条件关于矩阵的元素都是逐个可分的, 采用类似的方法可证明 $a = \frac{1}{2}$ 时的最优解为 $H_{\lambda}(W)$, 即以 λ 为阈值对矩阵 W 的每个元素进行 Half 阈值收缩, 相应的 Half 阈值算子 [11] 定义为

$$H_{\lambda}(x) = \begin{cases} \frac{2}{3}x \left(1 + \cos\left(\frac{2\pi}{3} - \frac{2\varphi(x)}{3}\right)\right), & |x| > \frac{\sqrt[3]{54}}{4}\lambda^{\frac{2}{3}}, \\ 0, & \text{其他}, \end{cases} \quad (14)$$

其中 $\varphi(x) = \arccos(\frac{\lambda}{8}(\frac{|x|}{3})^{-\frac{3}{2}})$.

问题 (12) 的求解相对困难, 我们以下述定理 1 给出它的显式解.

定理 1 记矩阵 $W \in \mathbb{R}^{m \times n}$ ($m \geq n$) 的秩为 $r \in [1, \min\{m, n\}]$, 其奇异值分解为 $W = U_{m \times r} D_{r \times r} \times V_{n \times r}^T$, 其中 $D = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_r)$, $U = [u_1, u_2, \dots, u_r]$, $V = [v_1, v_2, \dots, v_r]$, $u_i \in \mathbb{R}^m$, $v_i \in \mathbb{R}^n$, 则问题 (12) 的显式解为 $X^* = UH_{\lambda}(D)V^T$.

证明 记决策变量 X 的奇异值分解为

$$X = U'D'V'^T, \quad (15)$$

其中 $D' = \text{diag}(\sigma'_1, \sigma'_2, \dots, \sigma'_n)$, $U' = [u'_1, u'_2, \dots, u'_n]$, $V' = [v'_1, v'_2, \dots, v'_n]$.

将非凸优化问题 (12) 的目标函数展开可得

$$\|W - X\|_F^2 + \lambda \|X\|_{S_{1/2}}^{1/2} = \|W\|_F^2 - 2 \sum_{i=1}^n \sigma'_i u_i'^T W v_i' + \sum_{i=1}^n \sigma_i'^2 + \lambda \sum_{i=1}^n \sqrt{\sigma_i'}. \quad (16)$$

令 $Q(U', V') = \min_{D' \geq 0} -2 \sum_{i=1}^n \sigma'_i u_i'^T W v_i' + \sum_{i=1}^n \sigma_i'^2 + \lambda \sum_{i=1}^n \sqrt{\sigma_i'}$, 则问题 (12) 等价于

$$\min_{U', V'} Q(U', V') \quad \text{s.t.} \quad U'^T U' = I_n, \quad V'^T V' = I_n. \quad (17)$$

记 $t_i = u_i'^T W v_i'$, $f_\lambda(\sigma'_i) = -2\sigma'_i t_i + \sigma_i'^2 + \lambda \sqrt{\sigma_i'}$, 则

$$Q(U', V') = \min_{D' \geq 0} \sum_{i=1}^n f_\lambda(\sigma'_i). \quad (18)$$

由于 $Q(U', V')$ 关于 D' 的对角元可分, 求解问题 (18) 等价于对所有的 i ($1 \leq i \leq n$) 求解

$$\sigma_i^* = \arg \min_{\sigma'_i \geq 0} f_\lambda(\sigma'_i). \quad (19)$$

参考文献 [11] 中有关向量 Half 阈值算子的推导过程可得 $\sigma_i^* = H_\lambda(t_i)$, 且满足

$$-2t_i + 2\sigma_i^* + \frac{\lambda}{2\sqrt{\sigma_i^*}} = 0. \quad (20)$$

记 $g(t_i)$ 为 $f_\lambda(\sigma_i^*)$ 关于 t_i 的导函数, 则有

$$g(t_i) = (-2t_i + 2\sigma_i^* + \frac{\lambda}{2\sqrt{\sigma_i^*}}) \frac{d\sigma_i^*}{dt_i} - 2\sigma_i^*. \quad (21)$$

将等式 (20) 代入上式得到 $g(t_i) = -2\sigma_i^* < 0$, 这说明 $f_\lambda(\sigma_i^*)$ 为关于 $t_i = u_i'^T W v_i'$ 的单调减函数, 此时问题 (17) 的求解等价于求解

$$\max_{u_i', v_i'} u_i'^T W v_i' \quad \text{s.t.} \quad \|u_i'\|_2^2 = 1, \|v_i'\|_2^2 = 1, \quad u_i' \perp \{u_1^*, u_2^*, \dots, u_{i-1}^*\}, v_i' \perp \{v_1^*, v_2^*, \dots, v_{i-1}^*\}. \quad (22)$$

显然, 上述问题的最优解 u_i^* , v_i^* 分别等于矩阵 W 的第 i 个奇异值对应的左、右奇异向量 u_i , v_i , 目标函数的最优值为 $t_i^* = u_i'^T W v_i' = \sigma_i$. 若用 $r(\lambda)$ 表示矩阵 W 的大于 $\frac{\sqrt[3]{54}}{3} \lambda^{\frac{2}{3}}$ 的奇异值个数, 则问题 (17) 的最优解为 $[u_1, \dots, u_{r(\lambda)}]$ 与 $[v_1, \dots, v_{r(\lambda)}]$, 即矩阵 W 前 $r(\lambda)$ 个最大奇异值对应的左右奇异向量. 另外, 根据问题 (19) 的求解可得到最优的 $D' = H_\lambda(D)$.

综上, 由于矩阵 W 的秩为 r , 则问题 (12) 的最优解为 $X^* = U H_\lambda(D) V^T$. 证毕.

2.3 交替阈值迭代算法

利用问题 (11) 与 (12) 的显式解, 我们给出实现求解问题 (4) 的交替阈值迭代算法如下:

- 1) 初始化 $\{Y_0, E_0, A_0\}$, 选取适当的参数 μ , 设置误差精度 ε , 令 $k = 0$;
- 2) 更新低秩成分 A_{k+1} : $D - E_k + \mu^{-1} Y_k \approx U_r S_r V_r^T$, $A_{k+1} = U_r H_{2/\mu}(S_r) V_r^T$;
- 3) 更新稀疏成分 E_{k+1} :

$$E_{k+1} = \begin{cases} H_{2\lambda/\mu}(D - A_{k+1} + \mu^{-1} Y_k), & a = \frac{1}{2}, \\ ST_{\lambda/\mu}(D - A_{k+1} + \mu^{-1} Y_k), & a = 1; \end{cases} \quad (23)$$

4) 更新 Lagrange 乘子: $Y_{k+1} = Y_k + \mu(D - A_{k+1} - Y_{k+1})$;

5) 判断 $\frac{\|D - A_{k+1} - E_{k+1}\|_F}{\|D\|_F} < \varepsilon$ 是否成立, 是则终止迭代; 否则令 $k = k + 1$, 转步骤 2)。

由于 ADMM 框架下算法的收敛速度与参数 μ 的取值有关, 为避免算法收敛速度对参数 μ 初始值选取的依赖性并同时加速其收敛, 我们可在每次迭代中使用不同的参数^[14]。一种最简单的处理方式是每次迭代都对参数 μ 进行一次更新, 即取 $\mu_{k+1} = \rho\mu_k$, 其中 ρ 为大于 1 的常数。注意到更新低秩矩阵 A 用到的阈值仅与 μ_k 有关, 在预估计秩为 r 的情况下, 可利用 Half 阈值算子的特性如下自适应选取阈值^[11]:

$$\mu_{k+1} = \max \left\{ \mu_k, \frac{\sqrt{54}(\sigma_{r+1}^{(k)})^{-3/2}}{4} \right\}, \quad (24)$$

这里 $\sigma_{r+1}^{(k)}$ 为矩阵 $D - E_k + \frac{1}{\mu_k} Y_k$ 的第 $r+1$ 个奇异值。

为便于后文叙述, 如下约定不同参数取值与更新方式下的算法: $a = \frac{1}{2}$ 时, 若参数 μ_k 自适应更新, 简记为 AHH 算法 (adaptive half-half iterative alternating thresholding algorithm), 否则简记为 IHH 算法 (inadaptive half-half iterative alternating thresholding algorithm); $a = 1$ 且参数 μ_k 自适应更新时, 简记为 AHO 算法 (adaptive half-one iterative alternating thresholding algorithm)。

3 数值实验

本节将在有、无噪声干扰两种情形下, 比较 AHH, IHH, AHO 和 ALM 这 4 种算法实现 (近似) 稀疏-低秩矩阵分解的精度与效率, 并研究算法对低秩成分的低秩程度与稀疏成分的稀疏程度的稳健性。

对于待分解的矩阵 $D \in \mathbb{R}^{m \times m}$, 用矩阵中非零元素的比例 (记作 sr) 来刻画其稀疏成分 E 的稀疏程度, 用矩阵秩 r 与维数 m 的比值 (记作 lr) 来刻画其低秩成分 A 的低秩程度。我们采用与相关研究^[4,8,15] 相似的模式来生成矩阵 D , 具体步骤如下:

1) 独立生成随机矩阵 $L, R \in \mathbb{R}^{m \times r}$, 其中每个元素独立服从标准正态分布, 取低秩矩阵 $A = \frac{1}{\sqrt{r}} LR^T$ 以保证其每个元素的方差为 1;

2) 均匀随机选取稀疏矩阵 $E \in \mathbb{R}^{m \times m}$ 的 $sr \times m^2$ 个非零元的位置, 这些位置上的取值独立服从 $[0, 1]$ 区间上的均匀分布;

3) 生成各元素独立服从均值为 0 标准差为 σ 的正态分布的矩阵 $N \in \mathbb{R}^{m \times m}$, 令 $D = A + E + N$, $\sigma > 0$ 对应于有噪声干扰的情形。

由于 4 种算法都要预估计子空间矩阵的秩, 而模拟实验中真实秩是已知的, 我们通过大量的模拟实验发现: 当秩估计偏小时, AHH 与 AHO 算法不收敛; 当秩估计不小于真实秩也不超过真实秩的 10 倍时, 4 种算法都收敛且计算精度与迭代次数随秩估计的增大基本保持稳定, 但时间代价明显增加。因此, 模拟实验中 4 种算法的秩估计都取为真实秩的 1.5 倍。另外, 根据经验取模型 (4) 中的正则化参数 $\lambda = \frac{1}{\max\{m, n\}}$, 而 ALM 算法的参数设置始终如文献 [9] 所建议的那样取为 $\lambda = \frac{1}{\sqrt{\max\{m, n\}}}$ 。所有实验是在台式机 (CPU 为 2.30 GHz, 内存为 3.25 GB) 上使用软件 Matlab R2011a 完成的。

3.1 无噪声情形下的准确分解

当矩阵 D 不受噪声干扰, 可分解成严格低秩矩阵与严格稀疏矩阵的加和时, $S_{1/2}$ 模型 (4) 中的参数 $\delta = 0$, 我们将相应的 AHH 和 IHH 算法与 ALM 算法进行如下两方面的比较。

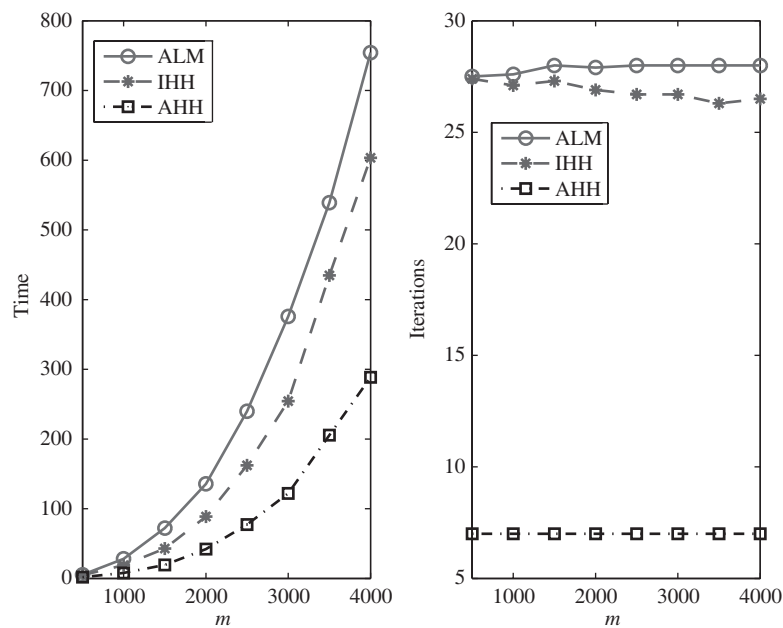


图 1 固定低秩程度 $lr=0.01$ 和稀疏程度 $sr=0.05$, IHH, AHH 与 ALM 三种算法实现不同规模矩阵分解时所需时间代价与迭代次数的比较

Figure 1 Comparison of time cost and iterations that IHH, AHH and ALM take when applied in matrix decomposition with low-rank ratio $lr = 0.01$, sparse ratio $sr = 0.05$ and varying size $m = 500 : 500 : 4000$

1) 计算精度与效率: 考虑不同大小矩阵 D ($m = 500 : 500 : 4000$), 令 $lr = 0.01$, $sr = 0.05$. 在每组参数设置下重复实验 20 次. 记算法的分解结果为 (A^*, E^*) , 图 1 和 2 与表 1 记录了 $\text{err}A^*$, $\text{err}E^*$, $\text{rank}(A^*)/\text{rank}(A)$ 与 $\|E^*\|_{l_0}/\|E\|_{l_0}$ 这 4 项指标的平均值来刻画算法的计算精度, 算法达到收敛所需迭代次数与时间代价的平均值来刻画算法的计算效率, 其中相对误差 $\text{err}A^*$ 与 $\text{err}E^*$ 的定义如下:

$$\text{err}A^* = \frac{\|A - A^*\|_F}{\|A\|_F}, \quad \text{err}E^* = \frac{\|E - E^*\|_F}{\|E\|_F}. \quad (25)$$

分析结果发现: 在计算精度方面, 3 种算法分解出的低秩成分与稀疏成分的相对误差都很小, 但 ALM 算法得到的低秩矩阵的秩略微偏大, IHH 与 AHH 算法能得到正确的秩, 且 AHH 算法分解出的稀疏矩阵的稀疏度更接近于真实值; 在计算速率方面, 随着问题规模的增大, ALM 和 IHH 算法的时间代价呈近似指数增长且迭代次数稳定在 27 附近, 而 AHH 算法的时间代价始终更小且增长相对平缓, 迭代次数稳定为 7 次.

2) 对低秩程度与稀疏程度的稳健性: 稀疏 - 低秩矩阵分解要求低秩成分 A 足够低秩同时稀疏成分 E 足够稀疏, 即 lr 和 sr 足够小, 下面将比较 3 种算法对这两个指标的稳健性.

我们对 $m = 1000$ 的矩阵设计了两组实验, 一组是固定 $sr = 0.05$ 而变化 $lr = 0.005 : 0.005 : 0.06$, 另一组则固定 $lr = 0.01$ 而变化 $sr = 0.03 : 0.03 : 0.6$. 图 3 和 4 分别记录了两组实验在不同参数设置下重复 20 次时, 3 种算法分解出低秩矩阵的秩与相对误差、迭代次数、时间代价这 4 项指标的平均值.

分析结果发现: 随着 lr 的增大, 3 种算法计算精度与迭代次数基本不变, 而 AHH 算法的时间代价与迭代次数始终少于 ALM 算法, 且总能得到正确的 $\text{rank}(A^*)$; 当 sr 分别增大到一定程度时 3 种算法均失效, $\text{rank}(A^*)$ 与真实值有很大的偏差, 其中 IHH 算法对 sr 的要求最低.

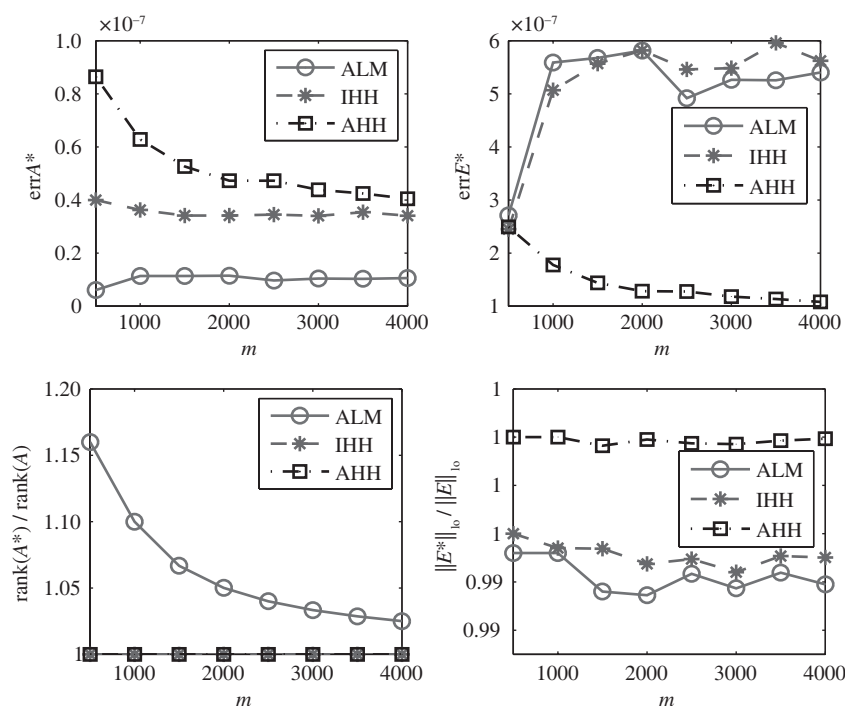


图 2 对不同规模的矩阵分解问题, 固定低秩程度 $lr=0.01$ 和稀疏程度 $sr=0.05$, IHH, AHH 与 ALM 三种算法分解结果的误差比较

Figure 2 Comparison of decomposition error when IHH, AHH and ALM are applied in matrix decomposition with low-rank ratio $lr = 0.01$, sparse ratio $sr = 0.05$ and varying size $m = 500 : 500 : 4000$

表 1 对不同规模的矩阵分解问题, AHH, IHH 和 ALM 三种算法计算精度与效率的比较 ($lr=0.01$, $sr=0.05$)

Table 1 Comparison of computational accuracy and efficiency when AHH, IHH and ALM are applied in matrix decomposition with varying size ($lr = 0.01$, $sr = 0.05$)

Performance	Algorithm	$m = 500$	$m = 1000$	$m = 1500$	$m = 2000$	$m = 2500$	$m = 3000$	$m = 3500$	$m = 4000$
rank(A^*)	ALM	5.8	11	16	21	26	31	36	41
	AHH	5	10	15	20	25	30	35	40
	IHH	5	10	15	20	25	30	35	40
Iterations	ALM	27.5	27.6	28	27.9	28	28	28	28
	AHH	7	7	7	7	7	7	7	7
	IHH	27.4	27.1	27.3	26.9	26.7	26.7	26.3	26.5
Time(s)	ALM	5.42	28.17	72.12	135.63	239.73	375.79	539.10	754.44
	AHH	1.82	7.86	19.30	42.03	77.35	121.91	205.56	288.60
	IHH	4.76	18.90	42.92	88.71	162.19	254.38	434.71	603.38

3.2 高斯噪声情形下的近似分解

考虑到上节中 AHH 算法表现出的高效性, 下面在有高斯噪声的情形下比较 AHO, AHH 和 ALM 这 3 种算法进行近似稀疏-低秩矩阵分解的能力.

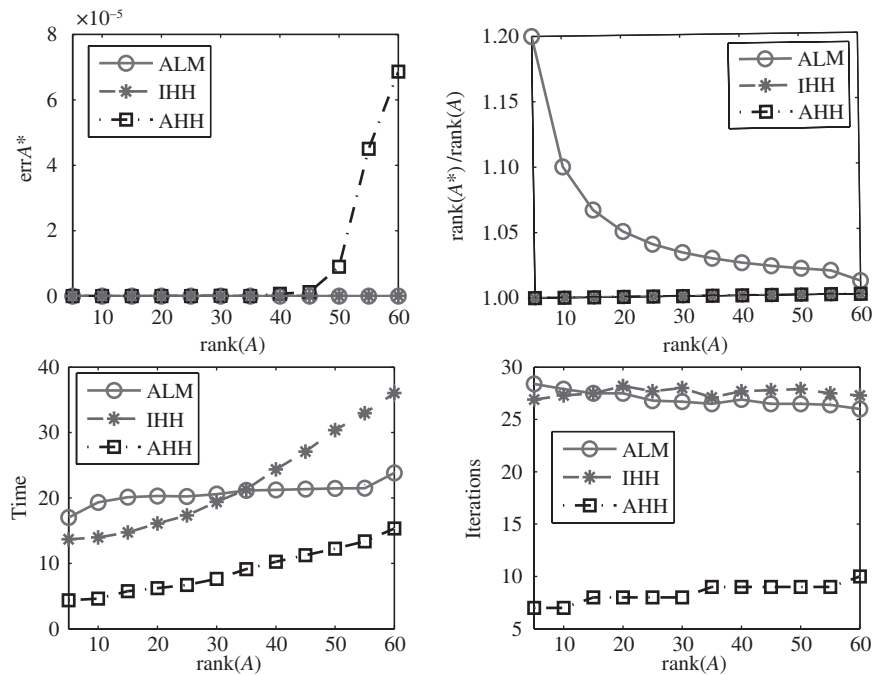


图3 对 $m=1000$ 的矩阵分解问题, 固定稀疏程度 $sr = 0.05$ 而变化低秩程度 $lr = 0.005 : 0.005 : 0.06$ 时, AHH, IHH 与 ALM 三种算法分解出的低秩矩阵的秩与相对误差、迭代次数和时间代价的比较

Figure 3 Comparison of the obtained low-rank matrix's rank and relative error, iterations and time cost when AHH, IHH and ALM are applied in the matrix decomposition with fixed $m = 1000, sr = 0.05$ and varying $lr = 0.005 : 0.005 : 0.06$

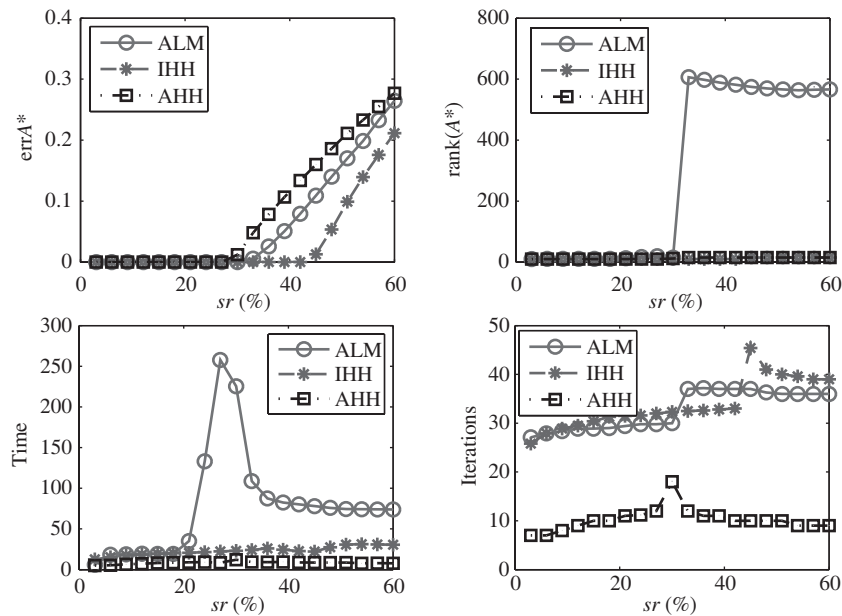


图4 对 $m=1000$ 的矩阵分解问题, 固定低秩程度 $lr=0.01$ 而变化稀疏程度 $sr=0.03 : 0.03 : 0.6$ 时, AHH, IHH 与 ALM 三种算法分解出的低秩矩阵的秩与相对误差、迭代次数和时间代价的比较

Figure 4 Comparison of the obtained low-rank matrix's rank and relative error, iterations and time cost when AHH, IHH and ALM are applied in the matrix decomposition with fixed $m = 1000, lr = 0.01$ and varying $sr = 0.03 : 0.03 : 0.6$

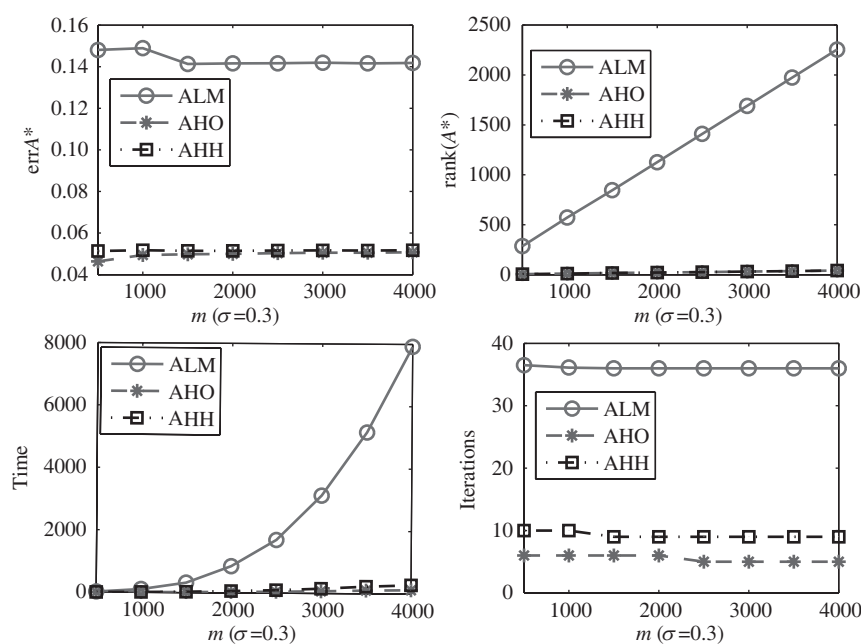


图 5 在有高斯噪声 $N(0, 0.3^2)$ 的情形下, 固定低秩程度 $lr=0.01$ 和稀疏程度 $sr=0.05$, AHO, AHH 与 ALM 三种算法实现不同规模矩阵分解时所需时间代价与迭代次数的比较

Figure 5 Comparison of the obtained low-rank matrix's rank and relative error, time cost and iterations when AHO, AHH and ALM are applied in matrix decomposition with low-rank ratio $lr=0.01$, sparse ratio $sr=0.05$ and varying size $m=500:500:4000$, in the case of Gaussian noise $N(0, 0.3^2)$

1) 计算精度与效率: 对不同大小的矩阵 D (变化 $m=500:500:4000$), 固定 $lr=0.01$ 与 $sr=0.05$, 变化高斯噪声的标准差 $\sigma=0.1:0.1:1$. 每一组参数设置下重复实验 10 次, 记录 3 种算法分解出的低秩矩阵的秩与相对误差、达到收敛所需迭代次数与时间代价这 4 项的平均值 (图 5 展示 $\sigma=0.3$ 时的结果). 分析结果发现: ALM 算法分解出的低秩矩阵的相对误差约为 AHO 和 AHH 算法的三倍, 而秩也远高于实际秩, 达到收敛所需迭代次数 (约 36 次) 是 AHO 算法 (约 6 次) 的 6 倍且时间代价随矩阵维数的增加呈指数增长趋势, 而 AHO 和 AHH 算法的时间代价则缓慢增加.

2) 对低秩程度与稀疏程度的稳健性: 对 $m=1000$ 的方阵, 取高斯噪声标准差 $\sigma=0.3$, 一方面固定 $sr=0.05$ 而变化 $lr=0.005:0.005:0.06$, 另一方面固定 $lr=0.01$ 而变化 $sr=0.03:0.03:0.6$. 每组参数设置下重复实验 20 次, 记录分解出的低秩成分的秩与相对误差、算法达到收敛所需的迭代次数与时间代价这 4 项指标的平均值 (图 6 和 7).

当固定 sr 而变化 lr 时, ALM 算法基本失效, $rank(A^*)$ 始终远大于 $rank(A)$, 达到收敛所需的迭代次数与时间代价也很大; AHO 与 AHH 算法则始终具有更强的稳健性, 恢复出的低秩矩阵的秩非常接近真实值, 两种算法达到收敛所需的迭代次数分别稳定在 7 次和 10 次, 且时间代价增长缓慢. 当固定 lr 而逐渐增加 sr 时, 3 种算法的恢复误差都逐渐增大而达到收敛所需的迭代次数与时间代价基本保持稳定, 但 AHO 与 AHH 算法的恢复误差始终小于 ALM 算法且恢复出的低秩矩阵的秩更接近于真实值, 同时迭代次数与时间代价更少 (详见表 2).

3) 对高斯噪声的稳健性: 为研究 3 种算法对高斯噪声的稳健性, 考虑 $m=1000$, $r=10$, $sr=0.05$ 的方阵, 变化高斯噪声标准差 $\sigma=0.05:0.05:1$. 对 σ 的每个取值重复实验 10 次, 表 3 记录了 3 种算法恢复出低秩矩阵的秩与相对误差和达到收敛所需迭代次数与时间代价的平均值.

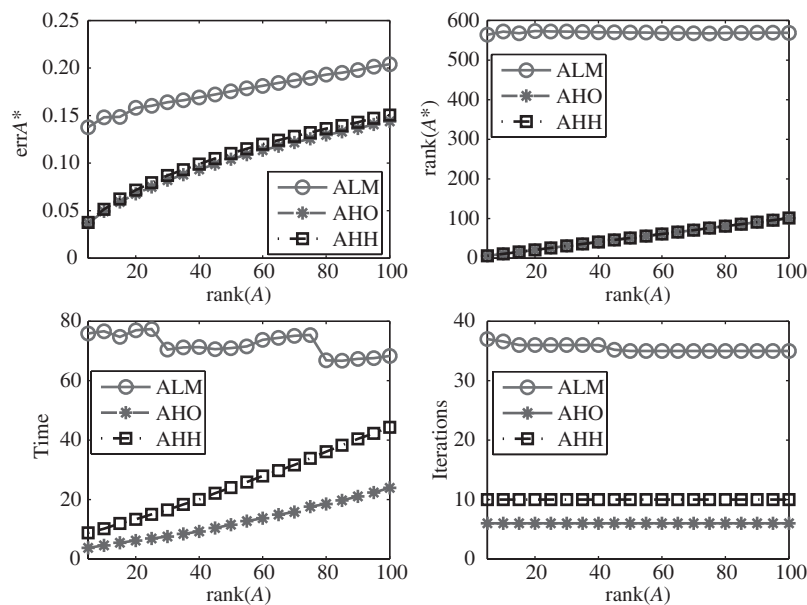


图6 对 $m=1000$ 且受高斯噪声 $N(0,0.3^2)$ 干扰的矩阵分解问题, 固定稀疏程度 $sr=0.05$ 而变化低秩程度 $lr=0.005:0.005:0.06$ 时, AHH, AHO 与 ALM 三种算法分解出的低秩矩阵的秩与相对误差、迭代次数和时间代价的比较

Figure 6 Comparison of the obtained low-rank matrix's rank and relative error, iterations and time cost when AHH, AHO and ALM are applied in the matrix decomposition with fixed $m=1000$, $sr=0.05$ and varying $lr=0.005:0.005:0.06$, in the case of Gaussian noise $N(0,0.3^2)$

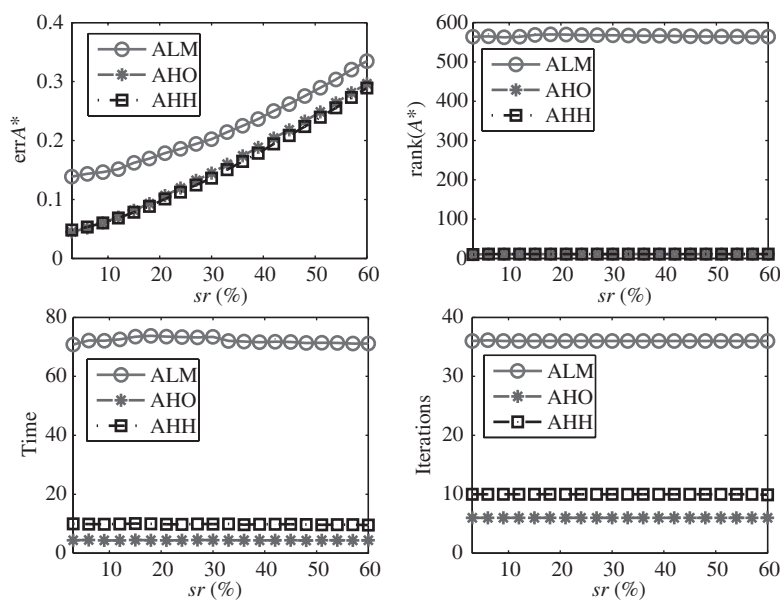


图7 对 $m=1000$ 且受高斯噪声 $N(0,0.3^2)$ 干扰的矩阵分解问题, 固定低秩程度 $lr=0.01$ 而变化稀疏程度 $sr=0.03:0.03:0.6$ 时, AHH, AHO 与 ALM 三种算法分解出的低秩矩阵的秩与相对误差、迭代次数和时间代价的比较

Figure 7 Comparison of the obtained low-rank matrix's rank and relative error, iterations and time cost when AHH, AHO and ALM are applied in the matrix decomposition with fixed $m=1000$, $sr=0.05$ and varying $lr=0.005:0.005:0.06$, in the case of Gaussian noise $N(0,0.3^2)$

表 2 AHH, AHO 和 ALM 三种算法对低秩程度与稀疏程度的稳健性 ($m=1000, \sigma=0.3$)Table 2 AHH, AHO and ALM's robustness to low-rank degree and sparsity for approximate decomposition ($m = 1000, \sigma = 0.3$)

Low-rank degree (lr)		0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10
err A^*	ALM	0.148	0.158	0.164	0.169	0.176	0.181	0.187	0.193	0.198	0.204
	AHO	0.049	0.067	0.082	0.093	0.104	0.114	0.122	0.130	0.137	0.144
	AHH	0.052	0.072	0.087	0.099	0.110	0.120	0.128	0.136	0.143	0.151
rank(A^*)	ALM	571	573	572	570	569	568	567	568	569	568
	AHO	11	21	31	41	51	61	71	81	91	101
	AHH	11	21	31	41	51	61	71	81	91	101
Sparsity(sr)		0.06	0.12	0.18	0.24	0.30	0.36	0.42	0.48	0.54	0.60
err A^*	ALM	0.144	0.152	0.169	0.186	0.202	0.225	0.250	0.275	0.304	0.335
	AHO	0.052	0.070	0.093	0.119	0.145	0.174	0.204	0.233	0.264	0.296
	AHH	0.053	0.069	0.088	0.112	0.136	0.165	0.194	0.224	0.255	0.289
rank(A^*)	ALM	565	564	570	568	567	566	566	565	565	564
	AHO	11	11	11	11	11	11	11	11	11	11
	AHH	11	11	11	11	11	11	11	11	11	11

表 3 AHH, AHO 和 ALM 三种算法对高斯噪声的稳健性 ($m=1000, lr=0.01, sr=0.05$)Table 3 AHH, AHO and ALM's robustness to Gaussian noise ($m = 1000, lr = 0.01, sr = 0.05$)

Indicator	Noise	$\sigma = 0$	$\sigma = 0.2$	$\sigma = 0.4$	$\sigma = 0.6$	$\sigma = 0.8$	$\sigma = 1$
err A^*	ALM	1.17E-8	0.095	0.190	0.279	0.355	0.455
	AHO	0.012	0.037	0.062	0.089	0.118	0.149
	AHH	5.46×10^{-8}	0.037	0.066	0.095	0.126	0.157
rank(A^*)	ALM	11	564.1	566.4	566.5	559.8	564.3
	AHO	11	11	10.6	10	10	10
	AHH	10	11	10.8	10	10	10
Iterations	ALM	27.5	36	36	36	35	35
	AHO	5	6	6	6	6	7
	AHH	7	10	10	10	10	10
Time(s)	ALM	18.2	73.26	70.75	71.05	68.68	68.75
	AHO	3.2	4.38	4.63	4.60	4.56	5.32
	AHH	5.00	10.11	9.83	10.03	9.76	9.89

分析结果发现: 3 种算法的 err A^* 与 σ 近似呈线性关系, 其中 AHH 与 AHO 算法对应直线的倾斜程度相近且远低于 ALM 算法; 3 种算法达到收敛所需迭代次数与时间代价基本稳定, 但 ALM 算法远高于 AHO 和 AHH 算法; ALM 算法在有噪声情形下分解不出低秩矩阵, 其 rank(A^*) 基本保持在 580. 另外, 值得注意的是, 在 $\sigma = 0$ 即无噪声的情形下, AHO 算法无法如同 AHH 算法一样恢复出正确的低秩矩阵, 这从一定程度上证实了本文对参数 a 启发式选取方法的合理性.

表 4 AHH, AHO, IHH 和 ALM 四种算法对数据集 Hall 和 Lobby 进行背景建模的结果比较

Table 4 Comparison of the resulting background modeling when AHH, AHO, IHH and ALM are applied in Hall and Lobby

Data set	Algorithm	ALM	AHO	AHH	IHH
Hall	rank(A^*)	134	6	5	9
	Iteration	37	9	8	29
	Time	108.7414	17.7623	33.2454	101.5765
Lobby	rank(A^*)	172	3	5	5
	Iteration	37	10	11	31
	Time	173.4526	16.4443	31.6590	88.3884

综合 3.1 与 3.2 两小节的实验结果可得到以下结论: 1) ALM 算法仅能较好地求解无噪声情形下的稀疏 - 低秩矩阵分解问题, 对噪声不具有稳健性, 分解出的低秩矩阵的秩远远高于真实秩. 2) 在无噪声的情形下, AHH 与 IHH 算法在很小的时间代价下就能达到与 ALM 算法相同的误差水平, 且得到的低秩成分的秩准确, 稀疏成分的非零元个数也更接近真实值. 3) 在有高斯噪声的情形下, AHO 与 AHH 算法具有更强的稳健性, 且计算效率也远高于 ALM 算法.

4 在视频监控中的应用

背景建模与监控视频的活动检测密切相关, 可自然地用稀疏 - 低秩矩阵分解模型进行建模. 若将视频的每一帧对应于待分解矩阵 D 的每一列, 则每帧中提取出的背景由于具有很强的相似性而构成低秩矩阵 A , 少量的运动目标和背景变化如光照等则对应于稀疏矩阵和噪声矩阵 $E + N$.

本节将分析文献 [4] 中的两段视频, 有显著前景变化的数据集 Hall 中每帧的大小是 176×144 , 有很多光照变化的数据集 Lobby 中每帧的大小是 168×120 . 由于每帧的背景非常相似, 所有帧的背景构成的矩阵应具有相当低秩的结构, 因此实验中 Hall 的背景矩阵秩预估计为 10, Lobby 的背景矩阵秩预估计为 5. 模型中参数 λ 的选取与前面模拟实验一致. 用 ALM, AHO, AHH, IHH 这 4 种算法对 250 帧 Lobby 与 300 帧 Hall 进行背景建模, 表 4 记录了各算法的时间代价、达到收敛所需迭代次数和构建的背景矩阵的秩, 图 8 展示了 AHO 算法对两个数据集进行背景建模后各两帧的效果, 其中第 1, 4 列对应于原始帧, 第 2, 5 列为相应的背景, 第 3, 6 列为相应的前景目标. (背景建模的 gif 格式文件可在主页 <http://gr.xjtu.edu.cn/web/zbxu/1> 下载).

从表 4 可看出, 在处理监控视频背景建模这一实际问题中, AHO 算法相较于其他算法有以下明显优势: 恢复出的背景矩阵更为低秩, 这一现象不仅符合建模时的先验, 同时为视频压缩与存储提供了新的可能; 达到收敛所需迭代次数很少, 时间代价相较于 ALM 算法下降量高达一个数量级, 这对于快速分析、处理海量视频数据有着重要意义.

5 结论

本文基于矩阵的 $S_{1/2}$ 范数对稀疏 - 低秩矩阵分解问题进行建模, 提出了能稳健、高效求解新模型的交替阈值迭代算法, 并将新模型与算法成功用于监控视频背景建模这一实际问题. 然而, 本文的



图 8 AHO 算法对数据集 Hall 和 Lobby 进行背景建模的结果 (第 1, 3 列为原始帧, 第 2, 4 列为稳定背景, 第 3, 6 列为运动前景)

Figure 8 Results of Background modeling when AHO algorithm is applied to Hall and Lobby (the first and third columns are the original frames, the second and fourth columns are the stable background, the third and sixth column are the moving foreground)

工作仍有两点需进一步完善: 第一个是低秩成分的秩估计问题, 希望能给出指导以帮助在实际应用中根据问题先验合理预估计低秩成分的秩; 第二个则是模型中参数 λ 的选取问题, 希望从理论上给出参数 λ 的选择策略. 显然, 前面两个问题是比较困难的, 目前已有的文献并没有给出让人较为满意的解答, 这将成为我们下一步工作研究的重点.

参考文献

- 1 Candès E J, Romberg J, Tao T. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Trans Inf Theory*, 2006, 52: 489–509
- 2 Donoho D L. Compressed sensing. *IEEE Trans Inf Theory*, 2006, 52: 1289–1306
- 3 Chandrasekaran V, Sanghavi S, Parrilo P A, et al. Rank-sparsity incoherence for matrix decomposition. Technical Report ArXiv: 0906.2220, 2009
- 4 Candès E J, Li X D, Ma Y, et al. Robust principle component analysis? *J ACM*, 2011, 58: article 11
- 5 Chandrasekaran V, Sangavi S, Parrilo P A, et al. Sparse and low-rank matrix decompositions. In: *Proceedings of IEEE 47th Annual Allerton Conference on Communication, Control, and Computing*, Illinois, 2009. 962–967
- 6 Wright J, Ganesh A, Rao S, et al. Robust PCA: exact recovery of corrupted low-rank matrices via convex optimization. In: *Advances in Neural Information Processing Systems*, Vancouver, 2009
- 7 Zhou Z, Li X, Wright J, et al. Stable principal component pursuit. In: *Proceedings of IEEE ISIT*, Austin, 2010. 1518–1522
- 8 Lin Z C, Ganesh A, Wright J, et al. Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix. *UIUC Technical Report UILU-ENG-09-2214*, 2009
- 9 Lin Z C, Chen M M, Ma Y. The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices. *UTUC Technical Report UILU-ENG-09-2214*, 2010
- 10 Xu Z B, Zhang H, Wang Y, et al. $L_{1/2}$ regularization. *Sci China Inf Sci*, 2010, 53: 1159–1169
- 11 Xu Z B, Chang X Y, Xu F M, et al. $L_{1/2}$ regularization: A thresholding representation theory and a fast solver. *IEEE Trans Neural Netw Learn Syst*, 2012, 23: 1013–1027
- 12 Boyd S, Parikh N, Chu E, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundat Trend Mach Learn*, 2010, 3: 1–122
- 13 Daubechies I, Defrise M, Christine D M. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Commun Pure Appl Math*, 2004, 57: 1413–1457
- 14 Rockafellar R T. Monotone operators and the proximal point algorithm. *SIAM J Control Optim*, 1976, 14: 877

- 15 Mackey L, Talwalkar A, Jordan M I. Divide-and-conquer Matrix Factorization. Technical Report arXiv: 1107.0789, 2011

Robust sparse and low-rank matrix decomposition based on $S_{1/2}$ modeling

RAO Guo^{1,2}, PENG Yi^{1,2} & XU ZongBen^{1,2*}

1 Institute for Information and System Science, Xi'an Jiaotong University, Xi'an 710049, China;

2 Ministry of Education Key Laboratory for Intelligent Networks and Network Security, Xi'an Jiaotong University, Xi'an 710049, China

*E-mail: zbxu@mail.xjtu.edu.cn

Abstract This paper introduces the $S_{1/2}$ -norm for matrices to induce their lower rank, based on which a new model for robust sparse and low-rank matrix decomposition is proposed. To the best of our knowledge, this is the first time that the $S_{1/2}$ -norm for matrices is used to characterize the low-rank property. Inspired by the alternating direction method of multipliers, we propose a computationally efficient algorithm, the alternating threshold iterative algorithm, for the new model. The proposed algorithm adopts the augmented Lagrange multiplier technique and iteratively updates both the low-rank and sparse components in explicit form, making the global computation accuracy and time cost controllable. Numerous numerical simulation experiments are presented to show that the new algorithm requires much less computation time to obtain a more robust decomposition. In addition, the rank of the low-rank component and sparsity of the obtained sparse matrix are much closer to their true values compared with those of the state-of-the-art algorithm, inexact ALM, in solving these problems. When applied to a background modeling application for surveillance video, the new algorithm recovers the background matrix with a lower rank, which is sort of consistent with the prior while modeling, while the time cost is only 10% of that of the inexact ALM algorithm. All these findings confirm that the new model and algorithm can solve practical problems more effectively and efficiently.

Keywords $S_{1/2}$ -norm, sparse and low-rank matrix decomposition, efficient and robust, alternative thresholding iteration



RAO Guo was born in 1988. She received a bachelor's degree in information and computer science from Xi'an Jiaotong University, Xi'an, in 2010. Currently, she is a postgraduate at Xi'an Jiaotong University. Her major is applied mathematics, while her research interests include machine learning and data mining.



PENG Yi was born in 1988. He received a bachelor's degree in applied mathematics from Xi'an Jiaotong University, Xi'an, in 2010. Currently, he is a postgraduate at Xi'an Jiaotong University. His major is applied mathematics, while his research interests include tensor factorization and image processing.



XU ZongBen was born in 1955. He received his Ph.D. in mathematics from Xi'an Jiaotong University in 1987. He now serves as Vice President of Xi'an Jiaotong University, Chief Scientist of the National Basic Research Program of China (973 Project), and Director of the Institute for Information and System Sciences at the university. He received the National Natural Science Award of China in 2007 and the CSIAM

Su Buchin Applied Mathematics Prize in 2008. He delivered a 45 minute talk at the International Congress of Mathematicians in 2010. His current research interests include intelligent information processing and applied mathematics. He is an Academician at the Chinese Academy of Sciences.