

基于矩阵低秩分解的图像修复

袁雨 001139.01 PB20151804

2023 年 4 月 9 日

摘要

某些情况下, 矩阵会呈现一定的低秩特性, 例如一系列从属于同一个人脸的图像, 又或者是具有一定周期规律特性的系列图片构成的矩阵。低秩分解是一种有效的图像修复方法, 它将破损的图像分解成矩阵的低秩部分和稀疏部分。本文旨在探讨基于矩阵低秩分解的图像修复问题。

本文主要介绍了鲁棒性主成分分析 (Robust Principle Component Analysis, RPCA) 问题及其求解算法。对于较大规模的数据, 经典的内点法计算复杂度较高。因此, 本文介绍了: 迭代阈值算法 (Iterative Thresholding, IT) 也叫作奇异值阈值法 (Singular Value Thresholding, SVT)、加速近端梯度算法 (Accelerated Proximal Gradient, APG)、增广的拉格朗日乘子法 (Augmented Lagrange Multiplier, ALM) 和非精确的增广拉格朗日乘子法 (Inexact Augmented Lagrange Multiplier, IALM) 也被称为交替方向算法 (Alternating Direction Method, ADM) 等, 这些算法可以用于解决 RPCA 模型在矩阵维度较大时求解困难的问题。

本文的实验结果表明, 上述的求解算法可以有效地解决 RPCA 算法求解困难的问题。同时本文对这些算法的效率进行了对比。

一、前言（问题的提出）

随着科技的快速发展，人们在生活中生产并消费着大量的数据，例如随手拍摄的照片、工作中收发邮件和越来越普及的指纹识别和人脸识别等。因此，对这些海量的数据，如何快速和高效地提取出其中的有效信息，已经成为数据挖掘和图像处理等研究领域的热点。

现代的数据通常具有很高的维度，我们如果直接对这些高维数据进行处理，会遇到很大的困难。首先，高维度的大规模数据会极大地提高数据的储存和计算成本，甚至导致高复杂度的算法在实际中无法应用；其次，这些数据在采集时不可避免的会出现噪点和离群点，例如在人脸识别中，可能会出现遮挡、光照和阴影等，因此会对处理算法的性能造成影响；此外，当高维空间的样本量较少时，这些数据会表现为稀疏的性质，例如推荐系统中的数据矩阵是典型稀疏的，我们很难从中挖掘出数据的内在信息。幸运的是，这些高维数据并不是互相独立的，而是具有一定的结构，并且可以用其线性子空间来描述。因此，在对高维数据进行处理前，通常对其进行维数约减，即从高维数据中挖掘出它所在的低维结构子空间。传统的维数约减方法，如主成分分析 (Principal Component Analysis, PCA) 和线性判别分析 (Linear Discriminant Analysis, LDA) 等子空间学习模型，当数据是理想的或者只含有高斯噪声时，能有效的提取出数据中的结构信息。但是如上所述，现实环境中存在大量含有噪声和离群点的高维数据，此时这些方法的效果往往不理想。

压缩感知理论指出，对于高维信号，使用一定的方式将其变换到低维空间，如果变换后的信号是稀疏的，使用 l_1 范数代替 l_0 范数进行优化，能够以极高的概率恢复出原始信号。由于压缩感知在理论和应用中的巨大成功，稀疏性成为描述数据的重要特征。与传统的子空间学习模型相比，基于稀疏和低秩约束的子空间模型在处理高维数据时表现出明显的优势。目前关于稀疏和低秩的理论仍在发展阶段，对该理论进行学习和研究，对于扩充子空间学习模型具有重要的意义。鲁棒主成分分析 (Robust Principal Component Analysis, RPCA) 是基于稀疏与低秩约束的代表模型，对其进行深入研究，有利于理解稀疏与低秩的本质，并反过来推动相关理论的发展。

二、相关工作

随着图像处理的不断发展，基于矩阵低秩分解的图像修复研究已经受到国内外越来越多研究者的关注。在短短几年的时间里，国内与国外均呈现出大量的针对阵低秩分解的研究工作。

三、问题分析

某些情况下，矩阵会呈现一定的低秩特性，例如一系列从属于同一个人脸的图像，又或者是具有一定周期规律特性的系列图片构成的矩阵，这类矩阵往往具有低秩的特性。再比如一段连续视频帧序列，如果将其的每一帧数据按同一规则排列成一个列矢量，再将所有的列矢量组成一个矩阵，则该矩阵具有低秩特性；相关研究结果表明^[48]，具有低秩特性的矩阵由于其列矢量具有相似性，在图像处理和计算机视觉处理

中往往可以对该类矩阵进行一定的降维与优化。当具有低秩特性的矩阵受到噪声、遮挡等等影响时, 低秩特性就会遭到破坏。研究证明^[1], 对含有噪声等干扰的低秩矩阵,

$$X' = \min \text{rank}(X), \text{ s.t. } F(X) = b$$

其中, 约束 X 为低秩矩阵, F 表示将原始 $m \times n$ 的数据 X 映射到大小为 $k \times 1$ 的 X' 矩阵。矩阵的秩 $\text{rank}()$ 可以表示为数据矩阵的非零奇异值的个数, 所以矩阵秩的最小化问题其实是 ℓ_0 范数最小化问题由向量转变到矩阵。矩阵秩的优化问题的应用主要有鲁棒性主成分分析 (Robust Principle Component Analysis, RPCA) 问题、矩阵恢复 (Matrix Recovery) 问题和矩阵补全 (Matrix Completion) 问题。本文就以鲁棒主成分分析 (RPCA) 模型与其求解算法进行讨论。

四、建模的假设

RPCA 是基于稀疏和低秩约束的基础模型, 当原始数据中存在缺失时, RPCA 模型可以求解矩阵补全问题等。此外, RPCA 模型假设原始数据存在于一个单独的子空间中。

五、符号说明

表 1: 符号说明

符号	说明
D	数据矩阵
L	低秩矩阵
S	稀疏矩阵
Y	拉格朗日乘子
\mathcal{L}	拉格朗日函数
\mathcal{S}	软阈值算子
\mathcal{G}	奇异值阈值算子

六、数学模型建立

6.1 RPCA 模型

假设给定的原始数据矩阵 $D \in R^{m \times n}$ 具有低维的结构空间, 那么可以将矩阵 D 表示成两个矩阵之和, 即 $D = L + S$, 其中, 矩阵 $L \in R^{m \times n}$ 是低秩的。主成分分析 (PCA) 的目的是找到一个低秩矩阵 L ,

使其与矩阵 D 的误差 $S = D - L$ 最小, 我们可以建立如下优化问题:

$$\begin{aligned} \min_{L, S} \|S\|_F \\ \text{s.t. rank}(L) \leq r, D = L + S, \end{aligned} \quad (1)$$

其中, $\|S\|_F$ 表示矩阵 S 的 Frobenius 范数, r 表示矩阵 L 的最大秩。针对问题 (1), 我们对矩阵 D 进行奇异值分解, 然后保留前 r 个最大奇异值对应的特征空间, 即可得到问题的最优解。当矩阵 S 中的元素是小高斯噪声时, 上述方法能得到最优的低秩矩阵 L 。但是对于其他类型的噪声, PCA 的效果不好。在实验中, 如果矩阵 S 中存在较大的元素, 即使元素数目很少, 通过 PCA 得到的低秩矩阵 L 也会和真实的低秩结构存在较大的偏差。

鲁棒主成分分析 (RPCA) 对 PCA 失效的情况进行了改进, 假定矩阵 S 是稀疏的并且其元素值可以任意大, RPCA 问题可以描述如下: 给定原始矩阵 $D = L + S$, 其中 L 是低秩的, S 是稀疏的, 要求恢复矩阵 L 。最直接的思路是寻找秩最低的矩阵 L 和非零元素最少的矩阵 S , 我们可以建立如下双目标优化问题:

$$\begin{aligned} \min_{L, S} (\text{rank}(L), \|S\|_0) \\ \text{s.t. } D = L + S, \end{aligned} \quad (2)$$

其中 $\text{rank}(L)$ 表示矩阵 L 的秩函数, $\|S\|_0$ 表示矩阵 S 的 l_0 范数, 即矩阵 S 中的非零元素之和。引入平衡参数 $\lambda > 0$, 用来衡量 $\text{rank}(L)$ 和 $\|S\|_0$ 对目标函数的贡献率, 问题 (2) 可以转化为一个更简单的单目标优化问题:

$$\begin{aligned} \min_{L, S} \text{rank}(L) + \lambda \|S\|_0 \\ \text{s.t. } D = L + S. \end{aligned} \quad (3)$$

该优化问题是非凸的, 而且是 NP 难的。令 $\|L\|_*$ 表示矩阵 L 的核范数, 即矩阵 L 的奇异值之和, $\|S\|_1$ 表示矩阵 S 的 l_1 范数, 即矩阵 S 中各元素的绝对值之和。下面给出两个引理:

引理 1 ^{[2] [3]} 矩阵的 l_1 范数是矩阵 l_0 范数的凸包络 (convex envelop)。

引理 2 ^[4] 矩阵的核范数是矩阵秩函数的凸包。

根据引理 1 和引理 2, 我们可以对问题进行松弛, 从而得到如下的凸优化问题:

$$\begin{aligned} \min_{L, S} \|L\|_* + \lambda \|S\|_1 \\ \text{s.t. } D = L + S, \end{aligned} \quad (4)$$

一般地, 我们称优化模型为鲁棒主成分分析模型。Candès 等人 ^[5] 给出证明, 在自然概率模型下, 即在绝大多数情景下, 通过求解凸优化问题, 可以精确地恢复出低秩矩阵 L 和稀疏矩阵 S 。

6.2 RPCA 求解算法

模型 (4) 是一个凸优化问题, 因此可以将其转化为半定规划问题, 并使用经典的内点法进行求解。内点法只需很少的迭代次数就可以使其收敛, 但是对规模为 $m \times m$ 的矩阵, 内点法的计算复杂度高达 $O(m^6)$, 在稍大规模的数据中即无法应用。我们下面介绍几种不受维数制约的求解算法: 迭代阈值算法 (Iterative Thresholding, IT) 也叫作奇异值阈值法 (Singular Value Thresholding, SVT) ^[6]、加速近端梯度算法 (Accelerated Proximal Gradient, APG) ^[7]、增广的拉格朗日乘子法 (Augmented Lagrange Multiplier, ALM) 和非精确的增广拉格朗日乘子法 (Inexact Augmented Lagrange Multiplier, IALM) 也被称为交替方向算法 (Alternating Direction Method, ADM) ^[8]。本节将对其分别作简单介绍。

为了方便后文对 RPCA 和相关模型进行求解, 我们首先引出软阈值算子 (soft-thresholding operator):

$$\mathcal{S}_\varepsilon[x] = \begin{cases} x - \varepsilon, & \text{如果 } x > \varepsilon \\ x + \varepsilon, & \text{如果 } x < -\varepsilon \\ 0, & \text{其他,} \end{cases}$$

其中 $x \in \mathbb{R}$ 并且 $\varepsilon > 0$ 。此外, 通过对元素进行操作, 该算子可以推广到向量和矩阵中。然后给出如下引理:

引理 3 假设矩阵 W 的奇异值分解为 $W = U\Sigma V^T$, 其中 $\Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_r)$, r 为矩阵 W 的秩, 那么核范数极小化问题存在闭式解:

$$\mathcal{G}_\varepsilon[W] = U\mathcal{S}_\varepsilon[\Sigma]V^T = \arg \min_X \varepsilon \|X\|_* + \frac{1}{2} \|X - W\|_F^2$$

引理 4 矩阵的 l_1 范数极小化问题存在闭式解:

$$\mathcal{S}_\varepsilon[W] = \arg \min_X \varepsilon \|X\|_1 + \frac{1}{2} \|X - W\|_F^2$$

6.2.1 迭代阈值法 (IT)

将优化问题 (4) 进行正则化, 我们可以得到如下问题:

$$\begin{aligned} \min_{L, S} \quad & \|L\|_* + \lambda \|S\|_1 + \frac{\mu}{2} (\|L\|_F^2 + \|S\|_F^2) \\ \text{s.t.} \quad & D = L + S, \end{aligned} \tag{5}$$

其中, $\mu > 0$ 是一个足够小的参数, 使得问题 (5) 是光滑的。对问题 (5) 引入拉格朗日乘子 Y , 可以得到拉格朗日函数:

$$\mathcal{L}(L, S, Y) = \|L\|_* + \lambda \|S\|_1 + \frac{\mu}{2} (\|L\|_F^2 + \|S\|_F^2) + \langle Y, D - L - S \rangle, \tag{6}$$

对优化问题 (6), 我们交替更新矩阵 L 、 S 和 Y 。当固定 $S = S_k$ 和 $Y = Y_k$ 时,

$$\begin{aligned} L_{k+1} &= \arg \min_L \mathcal{L}(L, S_k, Y_k) \\ &= \arg \min_L \mu^{-1} \|L\|_* + \frac{1}{2} \|L - \mu^{-1} Y_k\|_F^2 \\ &= \mathcal{G}_{\mu^{-1}}(\mu^{-1} Y_k) \\ &= \mu^{-1} \mathcal{G}_1(Y_k), \end{aligned}$$

当固定 $L = L_{k+1}$ 和 $Y = Y_k$ 时,

$$\begin{aligned} S_{k+1} &= \arg \min_S \mathcal{L}(L_{k+1}, S, Y_k) \\ &= \arg \min_S \lambda \mu^{-1} \|S\|_1 + \frac{1}{2} \|S - \mu^{-1} Y_k\|_F^2 \\ &= \mathcal{S}_{\mu^{-1}\lambda}(\mu^{-1} Y_k) \\ &= \mu^{-1} \mathcal{S}_\lambda(Y_k), \end{aligned}$$

当固定 $L = L_{k+1}$ 和 $S = S_{k+1}$ 时,

$$Y_{k+1} = Y_k + \delta_k (D - L_{k+1} - S_{k+1}),$$

其中, δ_k 表示步长, 且满足 $0 < \delta_k < 1$ 。

IT 算法的描述见表 2。

IT 算法的迭代方式十分简洁, 并且存在收敛性证明。但在实际应用中, 其收敛速度较慢, 难以适应规模较大的问题。

表 2: 迭代阈值法 (IT)

算法 1 迭代阈值法 (IT)
输入: 矩阵 $D \in R^{m \times n}$, 权重 λ 和 μ .
1: while not converged do
2: $(U, \Sigma, V) = \text{svd}(Y_{k-1});$
3: $L_k = U \mathcal{S}_\tau[\Sigma] V^T;$
4: $S_k = \mathcal{S}_{\lambda_\tau}[Y_{k-1}];$
5: $Y_k = Y_{k-1} + \delta_k (D - L_k - S_k);$
6: end while
输出: 矩阵 (L_k, S_k) .

6.2.2 加速近端梯度法 (APG)

我们可以将约束优化问题 (4) 转化为如下无约束优化问题:

$$\mathcal{L}(L, S, \mu) = \mu(\|L\|_* + \lambda\|S\|_1) + \frac{1}{2}\|D - L - S\|_F^2, \quad (7)$$

对问题 (7), 使用加速近端梯度法进行求解。记 $g(L, S, \mu) = \mu(\|L\|_* + \lambda\|S\|_1)$, $f(L, S) = \frac{\|D - L - S\|_F^2}{2}$, 于是 $\mathcal{L}(L, S, \mu) = g(L, S, \mu) + f(L, S)$ 。函数 $g(L, S, \mu)$ 不可微, $f(L, S)$ 是光滑的和利普希茨连续的, 即存在 $\mathcal{L}_f > 0$, 使得

$$\|\nabla f(L, S) - \nabla f(\tilde{L}, \tilde{S})\|_F \leq \mathcal{L}_f \|(L - \tilde{L}, S - \tilde{S})\|_F,$$

其中 $\nabla f(L, S)$ 表示函数 $f(L, S)$ 关于矩阵变量 L 和 S 的 Frechet 梯度, 此处取 $\mathcal{L}_f = 2$ 。对于矩阵 Y_L 和 Y_S , 作 $\mathcal{L}(L, S, \mu)$ 的部分二次逼近:

$$\begin{aligned} Q(L, S, \mu, Y_L, Y_S) &= g(L, S, \mu) + f(Y_L, Y_S) + \langle \nabla f(Y_L, Y_S), (L - Y_L, S - Y_S) \rangle \\ &\quad + \frac{1}{2} \mathcal{L}_f \|(L - Y_L, S - Y_S)\|_F^2 \\ &= g(L, S, \mu) + f(Y_L, Y_S) + \langle Y_L - (D - Y_S), L - Y_L \rangle + \langle Y_S - (D - Y_L), S - Y_S \rangle \\ &\quad + \frac{1}{2} \mathcal{L}_f \|L - Y_L\|_F^2 + \frac{1}{2} \mathcal{L}_f \|S - Y_S\|_F^2, \end{aligned} \quad (8)$$

根据公式 (8) 可得, 当 $S = S_k, Y_L = Y_L^k, Y_S = Y_S^k$ 和 $\mu = \mu^k$ 时,

$$\begin{aligned} L_{k+1} &= \arg \min_L Q(L, S_k, \mu_k, Y_L^k, Y_S^k) \\ &= \arg \min_L \mu_k \|L\|_* + \langle Y_L^k - (D - Y_S^k), L \rangle + \frac{1}{2} \mathcal{L}_f \|L - Y_L^k\|_F^2 \\ &= \mathcal{S}_{\mu_k / \mathcal{L}_f} (Y_L^k + (D - Y_L^k - Y_S^k) / \mathcal{L}_f), \end{aligned}$$

类似的, 当 $L = L_{k+1}, Y_L = Y_L^k, Y_S = Y_S^k$ 和 $\mu = \mu^k$ 时,

$$\begin{aligned} S_{k+1} &= \arg \min_S Q(L_{k+1}, S, \mu_k, Y_L^k, Y_S^k) \\ &= \mathcal{S}_{\mathcal{L}^{-1}\mu_k}(Y_S^k + \mathcal{L}_f^{-1}(D - Y_L^k - Y_S^k)), \end{aligned}$$

为了确定更新 Y_L 和 Y_S 时的步长, 我们设置参数

$$t_{k+1} = \frac{1}{2} \left(1 + \sqrt{1 + 4t_k^2} \right),$$

于是得到 Y_L 和 Y_S 的迭代更新公式:

$$\begin{aligned} Y_L^{k+1} &= L_k + \frac{t_k - 1}{t_{k+1}} (L_k - L_{k+1}), \\ Y_S^{k+1} &= S_k + \frac{t_k - 1}{t_{k+1}} (S_k - S_{k+1}), \end{aligned}$$

μ 的迭代步骤为:

$$\mu_{k+1} = \max(\eta\mu_k, \tilde{\mu}),$$

其中 $\tilde{\mu} > 0, 0 < \eta < 1$ 。

APG 算法的描述见表 3。

APG 算法是 IT 算法的改进, 优化了 IT 算法的迭代次数, 从而提高了求解速度。

表 3: 加速近端梯度法 (APG)

算法 2 加速近端梯度法 (APG)
输入: 矩阵 $D \in R^{m \times n}, \lambda$.
1: 初始化变量.
2: while not converged do
3: $Y_k^L = L_k + \frac{t_{k-1}-1}{t_k} (L_k - L_{k-1}),$
$Y_k^S = S_k + \frac{t_{k-1}-1}{t_k} (S_k - S_{k-1});$
4: $G_k^L = Y_k^L - \frac{1}{2} (Y_k^L + Y_k^S - D);$
5: $(U, \Sigma, V) = \text{svd}(G_k^L), L_{k+1} = U \mathcal{S}_{\mu_k/2} [G_k^L] V^T;$
6: $G_k^S = Y_k^S - \frac{1}{2} (Y_k^L + Y_k^S - D);$
7: $S_{k+1} = \mathcal{S}_{\lambda\mu_k/2} [G_k^S];$
8: $t_{k+1} = \frac{1+\sqrt{4t_k^2+1}}{2}; \quad \mu_{k+1} = \max(\eta\mu_k, \bar{\mu});$
9: end while
输出: 矩阵 (L_k, S_k) .

6.2.3 增广拉格朗日乘子法 (ALM)

首先, 构造问题 (3) 的增广拉格朗日函数:

$$\mathcal{L}(L, S, Y, \mu) = \|L\|_* + \lambda \|S\|_1 + \langle Y, D - L - S \rangle + \frac{\mu}{2} \|D - L - S\|_F^2,$$

当 $Y = Y_k$ 和 $\mu = \mu_k$ 时, 使用交替迭代法求解优化问题 $\min_{L,S} \mathcal{L}(L, S, Y_k, \mu_k)$ 。使用精确拉格朗日乘子法时, 交替迭代矩阵 L 和 S , 直到满足条件为止。当 $S = S_k$ 时,

$$\begin{aligned} L_{k+1} &= \arg \min_L \mathcal{L}(L, S_k, Y_k, \mu_k) \\ &= \arg \min_L \|L\|_* + \mu_k \|L - (D - S_k + Y_k/\mu_k)\|_F^2 / 2 \\ &= \mathcal{S}_{1/\mu_k}(D - S_k + Y_k/\mu_k), \end{aligned}$$

然后根据得到的 L_{k+1} 更新矩阵 S :

$$\begin{aligned} S_{k+1} &= \arg \min_S \mathcal{L}(L_{k+1}, S, Y_k, \mu_k) \\ &= \arg \min_S \|S\|_1 + \mu_k \|S - (D - L_{k+1} + Y_k/\mu_k)\|_F^2 / 2 \\ &= \mathcal{S}_{\lambda/\mu_k}(D - L_{k+1} + Y_k/\mu_k), \end{aligned}$$

记 L_{k+1} 和 S_{k+1} 分别收敛于 L_{k+1}^* S_{k+1}^* , 则矩阵 Y 的更新公式为:

$$Y_{k+1} = Y_k + \mu_k (D - L_{k+1}^* - S_{k+1}^*),$$

最后更新参数 μ :

$$\mu_{k+1} = \begin{cases} \rho \mu_k, & \mu_k \|S_{k+1}^* - S_k^*\|_F / \|D\|_F < \varepsilon, \\ \mu_k, & \text{其他,} \end{cases},$$

其中 $\rho > 1$ 为常数, $\varepsilon > 0$ 为比较小的正数。

文献 [8] 指出如果 μ_k 按照几何速度增长, 并且 μ_k 的增长速度越快, 精确的增广拉格朗日乘子法的收敛速度就会越快。然而, 通过数值测量表明, 当 μ_k 的值太大时, 迭代的方法求解子问题 $(L_{k+1}^*, S_{k+1}^*) = \arg \min_{L,S} \mathcal{L}(L, S, Y_k^*, \mu_k)$ 的收敛速度就会减慢。造成 EALM 算法计算量大的原因是矩阵的 SVD 分解, 所以 μ_k 的选取就要尽量保证矩阵的 SVD 分解的次数能够尽量得少。

精确的增广拉格朗日乘子法的算法描述见表 4。

实践中不一定要得到 L_{k+1}^* 和 S_{k+1}^* 的精确解, 非精确的增广拉格朗日乘子法的提出就是通过更新 A_k 和 E_k 来获得 RPCA 问题的最优解代替精确解, 此时矩阵 L 和 S 的迭代公式是:

$$\begin{aligned} L_{k+1} &= \arg \min_L \mathcal{L}(L, S_k, Y_k, \mu_k) \\ &= \mathcal{S}_{1/\mu_k}(D - S_k + Y_k/\mu_k), \\ S_{k+1} &= \arg \min_S \mathcal{L}(L_{k+1}, S, Y_k, \mu_k) \\ &= \mathcal{S}_{1/\mu_k}(D - L_{k+1} + Y_k/\mu_k), \end{aligned}$$

非精确增广拉格朗日乘子法的算法描述见表 5。

表 4: 精确的增广拉格朗日乘子法 (EALM)

算法 3 精确的增广拉格朗日乘子法 (EALM)

输入: 观测矩阵 $D \in R^{m \times n}$, 参数 γ .

1. 初始化参数.
2. while not converged do
3. $L_{k+1}^0 = L_k, S_{k+1}^0 = S_k, j = 0$;
4. while not converged do
5. $(U, \Sigma, V) = \text{svd}(D - S_{k+1}^j + \mu_k^{-1} Y_k)$;
6. $L_{k+1}^{j+1} = U S_{\mu_k'}[\Sigma] V^T$;
7. $S_{k+1}^{j+1} = S_{\gamma \mu^{-1}}[D - L_{k+1} + \mu_k^{-1} Y_k]$;
8. end while
9. $Y_{k+1} = Y_k + \mu_k (D - A_{k+1} - E_{k+1})$;
10. 更新 μ_k 到 μ_{k+1} ;
11. end while

输出: 矩阵 (L_k, S_k) .

表 5: 非精确增广拉格朗日乘子法 (IALM)

算法 4 非精确增广拉格朗日乘子法 (IALM)

输入: 数据矩阵 $D \in R^{n \times n}$, 权重参数 λ .

- 1: 初始化参数.
- 2: while not converged do
- 3: $(U, \Sigma, V) = \text{svd}(D - S_k + \mu_k^{-1} Y_k)$;
- 4: $L_{k+1} = U S_{\mu_k^{-1}}[\Sigma] V^T$;
- 5: $S_{k+1} = S_{2+\mu_k^{-1}}[D - L_{k+1} + \mu_k^{-1} Y_k]$;
- 6: $Y_{k+1} = Y_k + \mu_k (D - L_{k+1} - S_{k+1})$;
- 7: end while

输出: 矩阵 (L_k, S_k) .

七、结果（与对比）

近年来, 许多的研究学者对这几种算法进行评价分析, 其中著名专家 Ma Yi 对这几种求解算法也做了一定的对比, 得出了如下的实验对比结果^[9], 如表所示:

表 6: 矩阵低秩稀疏分解求解算法对比

Algorithm	Rank of estimate	Relative error in estimate of A	Time(s)
Singular Value Thresholding	20	3.4×10^{-4}	877
Accelerated Proximal Gradient	20	2.0×10^{-5}	43
Exact ALM	20	7.6×10^{-8}	4
Inexact ALM	20	4.3×10^{-8}	2

由表 6 可以看出, 在对矩阵的秩有相同的估计时, IALM 算法相对于其它几种算法不仅有较小的相对误差, 而且在算法的效率上也是最高的。

7.3 结果展示

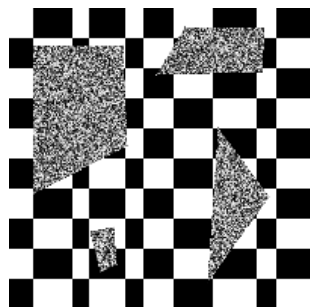


图 1: D1

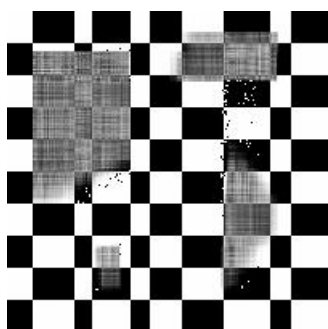


图 2: L1(APG)



图 3: S1(APG)

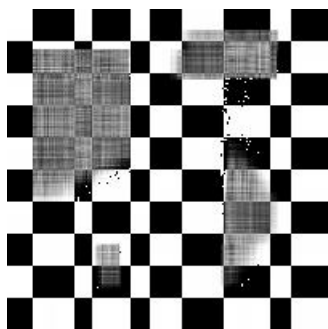


图 4: L1(EALM)



图 5: S1(EALM)

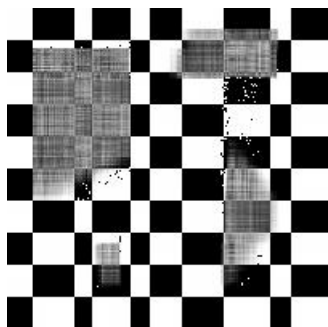


图 6: L1(IALM)



图 7: S1(IALM)

7.4 结果对比

IT 算法需要进行大量的迭代才能收敛，并且难以选择步长。故本文仅对 APG 算法、EALM 算法以及 IALM 算法进行展示。在图 1 上的实验结果如表 7 所示。

表 7: 实验结果

算法	迭代次数	耗时 (s)
<i>APG</i>	2000	18.73
<i>EALM</i>	10	13.55
<i>IALM</i>	1000	5.96

由图 2-7 可见这几种求解算法收敛的结果相似，算法较好得识别出了稀疏部分，但低秩部分恢复得并不好。

由表 7 知，收敛所花的时间排序是 $APG > EALM > IALM$ 。

八、结论

为了从受到强噪声的数据中恢复其低秩结构，Wright 等人^[7]提出了鲁棒主成分分析模型。近年来，该模型广泛应用于计算机视觉等研究领域，例如背景建模、人脸识别和图像处理等。它的主要思想是将原始数据矩阵分解为两个部分：低秩矩阵和稀疏矩阵，其中稀疏矩阵的元素取值可以任意大。本文详细

地介绍了鲁棒主成分分析的原理和相应的求解算法，包括迭代阈值法、加速近端梯度法和拉格朗日乘子法。其中，非精确增广拉格朗日乘子法也称为交替方向法，该方法具有精度高和收敛速度快的特点，已经成为求解稀疏与低秩模型的主流算法。

九、问题

非精确增广拉格朗日乘子法在求解鲁棒主成分分析模型时可以兼顾高效性和精确性。然而，在每次算法迭代中都要进行奇异值阈值操作。奇异值阈值算子需要通过奇异值分解求解。奇异值分解的计算量很大，是非精确增广拉格朗日乘子法的主要计算量。事实上，在奇异值阈值操作中，仅仅需要计算比特定的阈值大的奇异值。因此，可以考虑使用部分奇异值分解，这比完全奇异值分解更加高效。

本文仅使用了 RPCA 算法进行矩阵的低秩分解。以后有机会的话可以考虑使用其他算法。

本文在对比三种算法的性能时，仅使用了运行时间效率这一评价指标。以后有机会的话可以考虑使用结构相似度 (Structural Similarity Index of Measurement, SSIM)、峰值信噪比 (Peak Signal to Noise Ratio, PSNR) 等对算法进行对比。

参考文献

- [1] K. Mohan and M. Fazel, “New restricted isometry results for noisy low-rank recovery,” in *2010 IEEE International Symposium on Information Theory*, pp. 1573–1577, IEEE, 2010.
- [2] W. Yin, S. Osher, D. Goldfarb, and J. Darbon, “Bregman iterative algorithms for ℓ_1 -minimization with applications to compressed sensing,” *SIAM Journal on Imaging sciences*, vol. 1, no. 1, pp. 143–168, 2008.
- [3] W. Yin, “Augmented ℓ_1 and nuclear-norm models with globally linearly convergent algorithms,”
- [4] M. Fazel, *Matrix rank minimization with applications*. PhD thesis, PhD thesis, Stanford University, 2002.
- [5] E. J. Candès, X. Li, Y. Ma, and J. Wright, “Robust principal component analysis?,” *Journal of the ACM (JACM)*, vol. 58, no. 3, pp. 1–37, 2011.
- [6] J. Wright, A. Ganesh, S. R. Rao, and Y. Ma, “Robust principal component analysis: Exact recovery of corrupted low-rank matrices,” 2009.
- [7] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma, “Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization,” *Advances in neural information processing systems*, vol. 22, 2009.
- [8] Z. Lin, M. Chen, and Y. Ma, “The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices,” *arXiv preprint arXiv:1009.5055*, 2010.

[9] “http://perception.csl.illinois.edu/matrix-rank/sample_ode.html,”