

数学建模实验报告

实验二 基于矩阵分解的图像处理

姓名: 王世烜

学号: PB20151796

学院: 大数据学院

001139.01 数学建模
(春季, 2023)

中国科学技术大学

2023 年 4 月 3 日

摘 要

图像修复问题在信号处理、图像处理、机器学习等领域有着广泛的应用。**RPCA 问题**是指将一个矩阵分解为一个低秩矩阵和一个稀疏矩阵的问题。本文基于**矩阵低秩分解**的方法，复现了两种求解 RPCA 问题的算法：ADM 和 APG。

ADM 方法是一种基于交替方向乘子法的求解方法，**APG 方法**则是一种基于梯度下降的求解方法。通过对图像进行矩阵分解，将其分解为低秩矩阵和稀疏矩阵，从而实现对图像的修复。实验结果表明，算法能够有效地恢复图像的缺失部分，且具有较好的修复效果，不同的参数取值对于修复结果具有较大影响，应该视具体情况进行调整。

Abstract

Image restoration is widely used in signal processing, image processing, machine learning, and other fields. **RPCA problem** refers to the problem of decomposing a matrix into a low-rank matrix and a sparse matrix. Based on the matrix low-rank decomposition method, this paper reproduces two algorithms for solving the RPCA problem: ADM and APG.

ADM is a solution method based on alternating direction multiplier method, while **APG** is a solution method based on gradient descent. By decomposing the image into a low-rank matrix and a sparse matrix, the missing parts of the image can be restored. The experimental results show that the algorithm can effectively restore the missing parts of the image and has good restoration effect. Different parameter values have a significant impact on the restoration results, and should be adjusted according to specific situations.

目 录

1	前言	4
2	问题分析	4
3	建模的假设	4
4	符号说明	5
5	数学模型建立	5
5.1	加速近端梯度法 (Accelerated Proximal Gradient)	5
5.2	交替方向法 (Alternating Direction Methods)	7
6	结果 (与对比)	7
6.1	方法对比	7
6.2	参数对比	9
7	结论	10
8	问题	10
A	其他测试结果	12

1 前言

图像修复是计算机视觉领域的一个重要研究方向。在实际应用中，由于各种原因，图像可能会出现缺失、噪声等问题，这些问题会影响到图像的质量和可用性。因此，如何对图像进行修复，成为了一个热门的研究方向。本文旨在探究基于矩阵低秩分解的图像修复方法，通过实验验证其有效性。

2 问题分析

图像修复的目标是恢复图像中的缺失部分，使其能够更好地满足实际需求。在实际应用中，图像可能会出现各种问题，如噪声、失真、缺失等。其中，图像缺失是一种比较常见的问题。图像缺失可能是由于图像采集过程中的问题，也可能是由于图像传输过程中的问题。无论是哪种情况，都需要对图像进行修复，以便更好地满足实际需求。

3 建模的假设

我们考虑这样一个问题：一般数据矩阵 M 会包含结构信息，也包含噪声。那么我们可以将这个矩阵分解为两个矩阵相加，一个是低秩的（由于内部有一定的结构信息，造成各行或列间是线性相关的），另一个是稀疏的（由于含有噪声，而噪声是稀疏的），如图 1

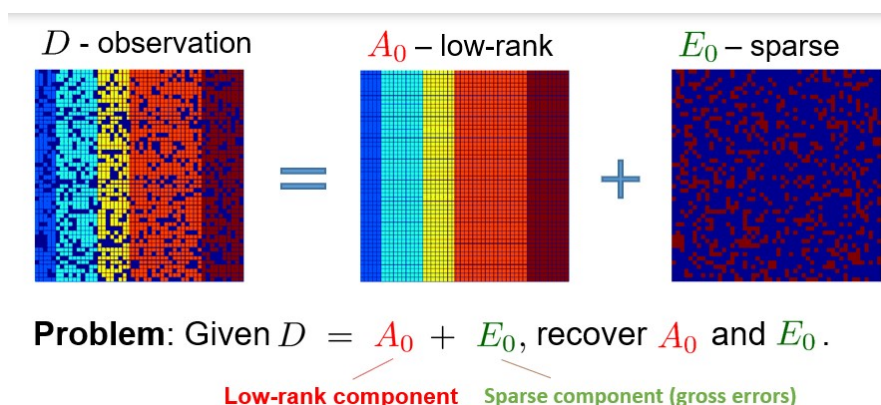


图 1

则图像修复可以写成以下的优化问题：

$$\begin{aligned} \arg \min_{L, S} \quad & \text{rank}(L) + \|S\|_0 \\ \text{s.t.} \quad & M = L + S \end{aligned} \quad (1)$$

其中 M 代表图像矩阵， L (Low Rank) 代表低秩矩阵， S (Sparse) 代表稀疏矩阵。 $\|S\|_0$ 代表矩阵的 0 范数，就是矩阵中非零元素的个数。但上述式子有一些弊端。虽然理论上可以计算，但是实际上不可行，因为这是一个 NP 问题，计算量很大。事实上，从已经证明的结论我们得知，可以用 1 范数来近似 0 范数，用核范数来近似矩阵的 rank 函数，可以得到以下问题：

$$\begin{aligned} \arg \min_{L, S} \quad & \|L\|^* + \lambda \|S\|_1 \\ \text{s.t.} \quad & M = L + S \end{aligned} \quad (2)$$

这个问题是一个凸优化问题，可以利用凸优化的方法求解。

4 符号说明

表 1: 符号说明

符号	含义	表达式
$\ A\ _0$	矩阵 A 的 0 范数	$\sum_{i,j} I(a_{ij} = 0)$
$\text{rank}(A)$	矩阵 A 的秩	—
$\ A\ _1$	矩阵 A 的 1 范数	$\sum_{i,j} a_{ij} = 0$
$\ A\ _*$	矩阵 A 的谱范数	$\sum_i \sigma_k(A), \sigma_k(A)$ 表示矩阵的第 k 个奇异值
$\ A\ _F$	矩阵 A 的 F 范数	$\sqrt{\sum_{i,j} A_{ij}^2}$
$S_\tau(X)$	shrinkage operator	$\text{sgn}(x) \max(x - \tau, 0)$
$D_\tau(X)$	singular value thresholding operator	$US_\tau(\Sigma)V^*$, 这里 $X = U\Sigma V^*$ 是 X 的奇异值分解
A^*	矩阵 A 的转置	—
$[A]_+$	矩阵 A 中正元素不变负元素变为 0	—

5 数学模型建立

前面提到，我们的问题可以转化为：

$$\begin{aligned} \arg \min_{L, S} \quad & \|L\|^* + \lambda \|S\|_1 \\ \text{s.t.} \quad & M = L + S \end{aligned}$$

我们分别采用两种凸优化算法对该问题进行迭代求解，分别为：**加速近端梯度法 (APG)** 和 **交替方向法 (ADM)**。

5.1 加速近端梯度法 (Accelerated Proximal Gradient)

在这篇论文 [1] 中作者使用加速近端梯度算法对于 (2) 式中的问题进行了松弛，加上了惩罚项，将其转化为如下新问题：

$$\arg \min_{L, S} L(L, S, \mu) = \mu(\|L\|^* + \lambda \|S\|_1) + \frac{1}{2} \|M - L - S\|_F^2 \quad (3)$$

记 $g(L, S, \mu) = \mu(\|L\|^* + \lambda \|S\|_1)$, $f(L, S) = \frac{1}{2} \|M - L - S\|_F^2$ ，可以证明 $g(L, S, \mu)$ 不可微而 $f(L, S)$ 是光滑的且其 Frechet 梯度 Lipschitz 连续，即 $\exists l_f > 0, \text{s.t. } \forall L, S, \bar{L}, \bar{S}$, 有

$$\|\nabla f(L, S) - \nabla f(\bar{L}, \bar{S})\|_F \leq l_f \|(\bar{L} - L, \bar{S} - S)\|_F$$

于是我们可以对于拉格朗日函数 $L(L, S, \mu)$ 进行部分二次逼近，其中 Y_L 和 Y_S 与 M 同型。

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

于是我们可以得到以下更新公式：

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

$$\begin{aligned}
S_{k+1} &= \arg \min_S Q(L_{k+1}, S, \mu_k, Y_L^k, Y_S^k) \\
&= S_{\mu_k/l_f}(Y_S^k + (M - Y_L^k - Y_S^k)/l_f)
\end{aligned}$$

$$t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2}$$

$$Y_L^{k+1} = L_k + (t_k - 1)(L_k - L_{k+1})/t_{k+1}$$

$$Y_S^{k+1} = S_k + (t_k - 1)(S_k - S_{k+1})/t_{k+1}$$

$$\mu_{k+1} = \max(\eta\mu_k, \hat{\mu}), \hat{\mu} \text{ is a constant}, 0 < \eta < 1$$

下面给出算法流程：

Algorithm 1 Robust PCA via Proximal Gradient with Continuation

Input: Observation matrix $M \in R^{m \times n}$, weight λ .

Output: L, S

1: $L_0, L_{-1} \leftarrow 0, S_0, S_{-1} \leftarrow 0, t_0, t_{-1} \leftarrow 1, \mu_0 \leftarrow 0.99\|D\|_{2,2}, \bar{\mu} \leftarrow 10^{-5}\mu_0$.

2: **while** Not Converged **do**

3: $\hat{L}_k = L_k + (t_k - 1)(L_k - L_{k+1})/t_{k+1}$

4: $\hat{S}_k = S_k + (t_k - 1)(S_k - S_{k+1})/t_{k+1}$

5: $Y_L^k = \hat{L}_k - \frac{1}{2}(\hat{L}_k + \hat{S}_k - M)$

6: $(U, S, V) = SVD(Y_L^k), L_{k+1} = U[S - \frac{\mu}{2}I]_+ V^*$

7: $Y_S^k = \hat{S}_k - \frac{1}{2}(\hat{L}_k + \hat{S}_k - M)$

8: $S_{k+1} = \text{sign}[Y_S^k] \circ [|Y_S^k| - \frac{\lambda\mu}{2}]_+$

9: $t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2}$

10: **end while**

11: **return** Outputs

5.2 交替方向法 (Alternating Direction Methods)

APG 算法的缺点是计算较慢, 为了改进这一算法, 论文 [2] 采用了非精确的增广拉格朗日法对式 (2) 进行了松弛, 得到增广拉格朗日函数:

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

然后对 L 和 S 交替求极小, 可得到迭代公式:

$$\begin{aligned} L_{k+1} &= \arg \min_L L(L, S_{k+1}, Y_k, \mu_k) \\ &= D_{1/\mu_k}(M - S_{k+1} + Y_k/\mu_k) \end{aligned}$$

$$\begin{aligned} S_{k+1} &= \arg \min_S L(L_{k+1}, S, Y_k, \mu_k) \\ &= D_{\lambda/\mu_k}(M - L_{k+1} + Y_k/\mu_k) \end{aligned}$$

给出算法流程:

Algorithm 2 Principal Component Pursuit by Alternating Directions

Input: Observation matrix $M \in R^{m \times n}$

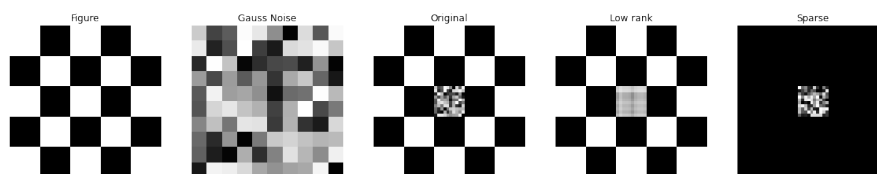
Output: L, S

- 1: $S_0 = Y_0 = 0, \mu > 0$
 - 2: **while** Not Converged **do**
 - 3: $L_{k+1} = D_{1/\mu_k}(M - S_{k+1} + Y_k/\mu_k)$
 - 4: $S_{k+1} = D_{\lambda/\mu_k}(M - L_{k+1} + Y_k/\mu_k)$
 - 5: $Y_{k+1} = Y_k + \mu(M - L_{k+1} + S_{k+1})$
 - 6: **end while**
 - 7: **return** Outputs
-

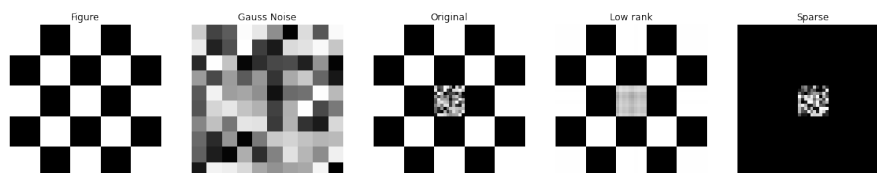
6 结果 (与对比)

6.1 方法对比

我们首先对一些自己生成的简单图片进行测试:

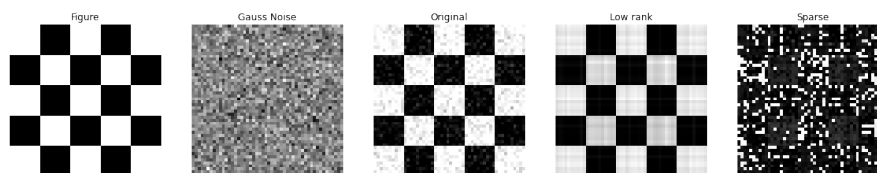


(a) APG, run-time=0.4s, Iteration=242

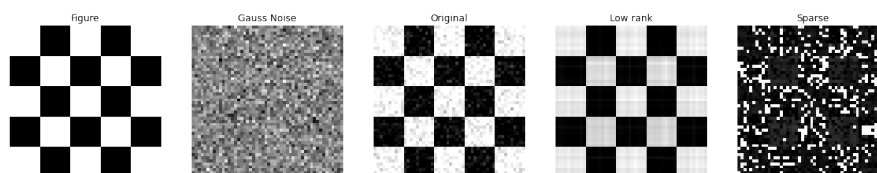


(b)ADM, run-time=0.2s,Iteration=54

这一图片是在原图的基础上只在最中心的一块加入了 $[0,255]$ 内的高斯随机噪声，从上面的图片可以看出，两个算法均较为出色的完成了任务，但实际上 ADM 算法的运行时间更短一些，收敛轮数更少一些，即收敛更快一些。



(c)APG, run-time=0.5s,Iteration=246



(d)ADM, run-time=0.3s,Iteration=135

这一张图片是对于整张图片随机添加均值为 0，标准差为 25 的高斯分布随机噪声。两种算法也是完成的比较好，同样的，ADM 算法收敛更快一些。

对于彩色图片，我们只需要分别对三个通道进行上述处理，也可以完成图像的恢复：

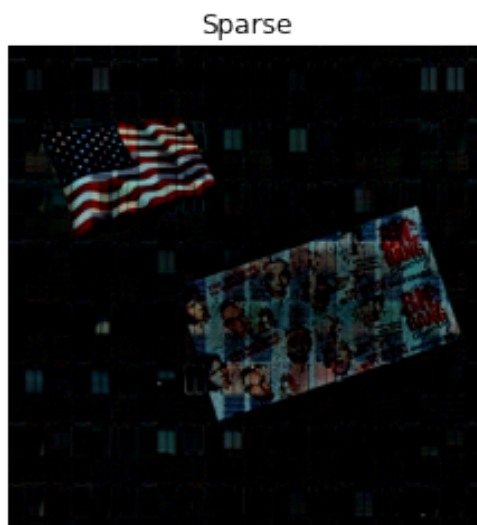


(e)APG, run-time=34.3s,Iteration=(266,265,265);ADM, run-time=12.6s,Iteration=(86,87,87)

这张图片的恢复效果很好。并且我们可以更明显的感受到 ADM 算法的时间性能提升。

上述图片的噪声情况效果并不是很好，对于图中的两块噪声，无法保持其原来的颜色，原因是因为如果将重心放在对于低秩图像的修复方面，则对于稀疏矩阵的非 0 部分，其与对应低秩矩阵的部分的和才是原图像中的非低秩（噪声）部分，这就是说低秩部分会对噪声部分的三通道灰度产生影响，无法得到真正的噪声。

但如果我们根据稀疏部分进行调参的话也可以得到可以接受的稀疏矩阵图像：



(f)Sparse

其他测试结果详见附录A.

6.2 参数对比

在以上的两种方法中均需要我们给定两个参数 λ 和 μ 。这两个参数有以下含义 (以 ADM 为例)：ADM 方法要优化的函数为：

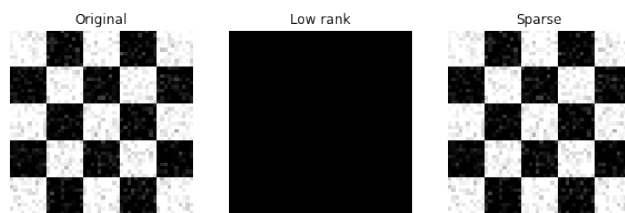
$$L(L, S, Y, \mu) = \|L\|^* + \lambda \|S\|_1 + \frac{\mu}{2} \|M - L - S\|_F^2 + \langle Y, M - L - S \rangle$$

参数 λ 代表对于 S 矩阵系数要求的严格程度，即当 S 不那么稀疏时受到的惩罚；参数 μ 代表对于矩阵 $L+S$ 与矩阵 M 的差距。在论文中作者分别给了参考初值 ($\lambda = \frac{1}{\sqrt{n}}, \mu = \frac{n_1 * n_2}{4 * \|M\|_1}$)，但实际上对于不同的图片，我们应该做出不同的调整。下面对于这参数 λ 分别进行调参并观察效果 (μ 参数同理本文不予讨论)。

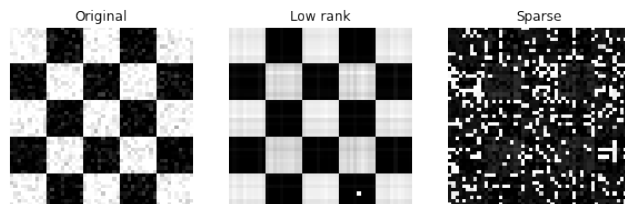
表 2: λ 与 rank(L)

$\lambda * \sqrt{n}$	0.1	0.2	0.3	0.4	0.5	1	1.5	10
rank(L)	0	0	0	0	1	2	7	50

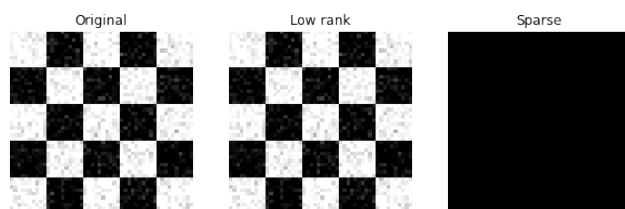
λ 越大，代表 S 稀疏占主导地位，则 rank(L) 的最小化要求就相对宽松，即可以取一些大于 0 的值，下面给出一些 λ 取值下的结果可以很清晰的感受到这个结论：



$\lambda = 0.1$



$\lambda = 1$



$\lambda = 10$

7 结论

本文介绍了 RPCA 问题及其求解方法，包括 ADM 和 APG 两种方法。实验结果表明，ADM 算法在时间性能上更优秀，而参数 λ, μ 的取值对结果有较大影响。在实际应用中，需要根据具体情况选择合适的方法和参数进行恢复。

8 问题

基于低秩分解进行图像修复算法对于**噪声的类型**比较敏感，从我的实验过程中可以感受到，对于分散的点噪声的处理效果较好，而对于连续的块噪声处理效果不是很好，有时会倾向于将噪声也处理成我们期望之外低秩的线，这样会对我们的结果造成较大影响。

除此之外，该算法对于大规模数据的处理能力较差，需要时间较长，如果待处理的对象变为视频时，通常的时间消耗是无法接受的。

参考文献

- [1] "Robust Principal Component Analysis: Exact Recovery of Corrupted Low-Rank Matrices via Convex Optimization", by Y. Bengio and D. Schuurmans and J. Lafferty and C. Williams and A. Cullotta, Curran Associates, Inc. 2009

- [2] Emmanuel J. Candes; Xiaodong Li; Yi Ma; John Wright (2009). "Robust Principal Component Analysis?". Journal of the ACM. 58 (3): 1–37.

附录 A 其他测试结果

