

# 基于线性 Bregman 迭代的结构化噪声矩阵补全算法

陈 蕾<sup>1),2)</sup> 杨 庚<sup>2)</sup> 陈正宇<sup>1)</sup> 肖 甫<sup>1)</sup> 陈松灿<sup>3)</sup>

<sup>1)</sup>(南京邮电大学计算机学院 南京 210003)

<sup>2)</sup>(南京邮电大学宽带无线通信与传感网技术教育部重点实验室 南京 210003)

<sup>3)</sup>(南京航空航天大学计算机学院 南京 210016)

**摘 要** 通过采样部分元素补全低秩矩阵的缺失元素是许多实际应用如图像修复、无线传感网数据收集和推荐系统等经常遇到的一个颇具挑战性的难题。在机器学习领域,这类问题通常能刻画成矩阵补全问题。虽然现有研究针对矩阵补全问题已提出了许多有效算法,但这些算法通常仅限于采样元素要么无噪要么仅含少量随机高斯噪声的补全情形,难以处理实际问题中常见的行结构化噪声。为了解决这个问题,该文首先借助分类器设计中流行的  $L_{2,1}$  范数正则化技术来平滑此类噪声,并将该问题建模为一类基于  $L_{2,1}$  范数正则化的凸约束优化问题。其次,为了快速有效地求解,我们将向量空间的线性 Bregman 迭代算法和近邻算子技术拓展到矩阵空间,进一步设计了一种鲁棒的基于线性 Bregman 迭代的结构化噪声矩阵补全算法(LiBIMC)。严格的理论分析证明了 LiBIMC 迭代算法的不动点正是结构化噪声矩阵补全问题的全局最优解。数值实验结果表明,和已有的矩阵补全算法相比, LiBIMC 算法不仅能更好地恢复结构化噪声矩阵的缺失元素,还能精确地辨识出采样矩阵中被污染的元素所在行的位置信息。

**关键词** 矩阵补全;结构化噪声;  $L_{2,1}$  范数正则化;线性 Bregman 迭代;近邻算子  
**中图法分类号** TP181; TP301 **DOI 号** 10.11897/SP.J.1016.2015.01357

## Linearized Bregman Iteration Algorithm for Matrix Completion with Structural Noise

CHEN Lei<sup>1),2)</sup> YANG Geng<sup>2)</sup> CHEN Zheng-Yu<sup>1)</sup> XIAO Fu<sup>1)</sup> CHEN Song-Can<sup>3)</sup>

<sup>1)</sup>(School of Computer, Nanjing University of Post and Telecommunications, Nanjing 210003)

<sup>2)</sup>(Key Laboratory of Broadband Wireless Communication and Sensor Network Technology of Ministry of Education, Nanjing University of Post and Telecommunications, Nanjing 210003)

<sup>3)</sup>(School of Computer, Nanjing University of Aeronautics and Astronautics, Nanjing 210016)

**Abstract** Recovering a low-rank matrix from incomplete samples of its entries is a challenging problem arising in many real world applications, such as image inpainting, data gathering for wireless sensor networks, and recommender system, etc. In machine learning community, this problem can be formulated as a general matrix completion problem. Recently, a lot of efficient algorithms have been proposed to solve this problem and they perform well when no noise or only Gaussian noise is added randomly to the sample data. However, when some number of the rows is arbitrarily corrupted, which named as the structural noise, the existing algorithms can return poor results. In order to smooth the structural noise, by introducing the  $L_{2,1}$ -norm regularization method, which is a popular trick in classifier design field, we formulate the matrix completion

收稿日期:2013-09-07;最终修改稿收到日期:2014-12-21。本课题得到国家“九七三”重点基础研究发展规划项目基金(2011CB302903)、国家自然科学基金(61272084,61201163,61272422,61373137)、江苏省自然科学基金(BK2011072, BK20130096)、江苏省高校自然科学研究重大项目(11KJA520002)及高等学校博士学科点专项科研基金(20113223110003)资助。陈 蕾,男,1975 年生,博士,副教授,中国计算机学会(CCF)会员,主要研究方向为机器学习、服务计算和信息安全。E-mail: chenlei@njupt.edu.cn。杨 庚,男,1961 年生,博士,教授,主要研究领域为网络安全、分布与并行计算、移动计算。陈正宇,男,1978 年生,博士研究生,副教授,主要研究方向为信息安全与隐私保护。肖 甫,男,1980 年生,博士,副教授,主要研究方向为无线传感器网络、多媒体图像检索等。陈松灿,男,1962 年生,博士,教授,主要研究领域为机器学习、模式识别与神经计算。

problem with structural noise as a convex constrained optimization problem based  $L_{2,1}$ -norm regularization. Then, in order to quickly and efficiently solve the convex constrained optimization problem, by generalizing the linearized Bregman iteration algorithm and the proximal operator method from vector space to the matrix space, we propose a robust linearized Bregman iteration algorithm called LiBIMC for matrix completion with structural noise. The theoretical analysis proves that the fixed point of the LiBIMC algorithm is exactly the global optimal solution of this matrix completion problem. Finally, numerical experiments show that, compared with other algorithms, the LiBIMC algorithm can better recover the matrix using correct measurements and detect the positions where the data is ruined.

**Keywords** matrix completion; structural noise;  $L_{2,1}$ -norm regularization; linearized Bregman iteration; proximal operator

## 1 引 言

近年来,压缩感知理论为信号采集技术带来了革命性的突破.众所周知,压缩感知理论要求在已知信号具有稀疏性的条件下对信号进行采集和重构,而在很多实际问题中,需要重构的目标常常是以矩阵的形式组织的.因此,压缩感知理论便自然地由向量空间被拓展至矩阵空间,从而利用矩阵奇异值向量的稀疏性,通过采样矩阵的部分元素来恢复低秩矩阵.在机器学习领域,这类问题通常被刻画为矩阵补全(Matrix Completion)问题.迄今为止,矩阵补全理论已在图像修复<sup>[1]</sup>、三维运动估计<sup>[2]</sup>、无线传感网数据收集<sup>[3]</sup>、多标记分类<sup>[4]</sup>、视频去噪<sup>[5]</sup>和推荐系统<sup>[6]</sup>等领域得到了重要应用,也逐渐成为机器学习、模式识别以及计算机视觉领域的主要研究热点之一.

现有研究针对矩阵补全问题已提出了许多有效算法.这些算法大致可分为两类:一类是基于核范数最小化的迭代算法,如 SVT 算法(Singular Value Thresholding)<sup>[7]</sup>、FPCA 算法(Fixed Point Continuation with Approximate SVD)<sup>[8]</sup>、IALM 算法(Inexact Augmented Lagrange Multipliers)<sup>[9]</sup>、ADM 算法(Alternating Direction Method)<sup>[10]</sup>、OptSpace 算法<sup>[11]</sup>和 MC\_TNNR 算法(Matrix Completion via Truncated Nuclear Norm Regularization)<sup>[12]</sup>等;另一类是基于矩阵分解的优化算法,包括 LMaFit 算法(Low-rank Matrix Fitting)<sup>[13]</sup>和 RTRMC 算法(Riemannian trust-region for MC)<sup>[14]</sup>等.这些矩阵补全算法在采样矩阵无噪或者仅含少量高斯随机噪声的情况下表现良好,但是当采样矩阵中的某些行受到损坏时,这些算法往往表现得不够稳定.本文将

采样矩阵中受到损坏的行称为结构化噪声,并将这种矩阵补全情形称为结构化噪声矩阵补全.

结构化噪声矩阵补全的一个典型应用场景是推荐系统.当前,随着互联网技术的日趋完善,网上购物等在线消费模式逐渐成为一种潮流,越来越多的人喜欢在网上购买自己需要的商品.对于商家而言,如何根据消费者反馈的不完全信息准确地预测他们的偏好从而给出合理的购物推荐成为了一个至关重要的问题.理想的情况是由消费者反馈的评分信息都是真实的,也就是说采样数据不含噪声.但是,实际上由于评分反馈过程缺乏有效的监督和约束机制,少部分消费者有时可能不愿意反馈他们的真实感受,从而随意给出他们的评分;更有甚至,个别不诚实的商家可能出于不当牟利的目的伪装或者串通消费者恶意地贬低竞争对手的产品评分同时抬高自己产品的评分.这样一来,如果我们将消费者作为行信息,产品作为列信息,就导致采样到的评分矩阵的某些行数据不真实,我们将其视为采样矩阵受到行结构化噪声污染.因此,在这种结构化噪声采样情形下如何补全评分矩阵的缺失元素,并识别那些噪声污染行元素的位置信息就显得尤为重要.

为了求解此类结构化噪声矩阵补全问题,本文引入了分类器设计中流行的  $L_{2,1}$  范数正则化惩罚因子来平滑结构化噪声,将该问题建模为一类基于  $L_{2,1}$  范数正则化的凸约束优化问题.为了进一步有效地求解该凸约束优化问题,本文将向量空间线性 Bregman 迭代算法推广到矩阵空间,采用矩阵空间近邻算子技术和线性 Bregman 迭代算法设计了一种鲁棒的基于线性 Bregman 迭代的结构化噪声矩阵补全算法(Linearized Bregman Iteration Algorithm for Matrix Completion with Structural Noise,

LiBIMC). LiBIMC 算法在求解的整个迭代过程都较好地保持了矩阵变量的低秩性和稀疏性, 因而使得该算法具有较高的执行效率和良好的可扩展性, 从而能适用于较大规模的矩阵补全问题. 同时, 严格的理论分析也证明了 LiBIMC 算法的不动点正是此类结构化噪声矩阵补全问题的全局最优解. 数值实验结果表明, 和 IALM、SVT 和 FPCA 算法相比, LiBIMC 算法不仅在没有移除噪声行的情况下获得了相对更好的补全精度, 更为重要的是, LiBIMC 算法能准确辨识出采样矩阵中被结构化噪声污染的行位置信息, 从而在补全不含噪部分缺失元素时可以事先移除采样矩阵的噪声行, 大大提高未受噪声污染行元素的补全精度. 从这个意义上来说, 在进行结构化噪声矩阵补全时, LiBIMC 算法可以作为现有矩阵补全算法的预处理步骤.

本文第 2 节对相关工作进行讨论; 第 3 节介绍一些数学预备知识; 第 4 节介绍向量空间线性 Bregman 迭代算法; 第 5 节为本文主要部分, 该部分针对结构化噪声矩阵补全问题提出了一类鲁棒的基于线性 Bregman 迭代的结构化噪声矩阵补全算法 (LiBIMC), 并从理论上证明 LiBIMC 算法的不动点即为结构化噪声矩阵补全问题的全局最优解; 第 6 节介绍一些数值实验并对实验结果进行分析; 第 7 节对所研究的内容进行总结并展望未来的研究方向.

## 2 相关工作

标准的矩阵补全问题最初被描述为

$$\min_{\mathbf{X} \in \mathbf{R}^{n_1 \times n_2}} \text{rank}(\mathbf{X}) \quad \text{s. t.} \quad P_\Omega(\mathbf{R}) = P_\Omega(\mathbf{X}) \quad (1)$$

其中  $\Omega \subseteq [n_1] \times [n_2]$  ( $[n_1] = \{1, \dots, n_1\}, n_2 = \{1, \dots, n_2\}$ ) 为采样元素的索引集合,  $P_\Omega(\cdot)$  为正交投影算子, 表示当  $(i, j) \in \Omega$  时,  $R_{ij}$  为采样元素, 即

$$[P_\Omega(\mathbf{R})]_{ij} = \begin{cases} R_{ij}, & (i, j) \in \Omega \\ 0, & (i, j) \notin \Omega \end{cases}$$

在式(1)中, 由于  $\text{rank}(\cdot)$  函数的非凸性和不连续性, 导致该问题求解成为 NP 难问题, 在所有已知的求解算法中, 其求解的复杂度均随着矩阵维数的增长呈平方倍指数关系增长. 类似于压缩感知理论中常用的将向量  $L_0$  范数松弛到  $L_1$  范数的技巧, 考虑到矩阵的秩实际上等同于该矩阵奇异值向量的  $L_0$  范数, 而矩阵的核范数则等同于奇异值向量的  $L_1$  范数. 因此, 为了使矩阵补全问题易于求解, 一个自然的想法就是利用凸的核范数代替非凸的秩函数, 将

问题(1)松弛为凸优化问题. 为此, Fazel 在文献[15]中严格证明了矩阵核范数是秩函数在矩阵谱范数意义下单位球上的最佳凸逼近. 随后, Recht 和 Fazel 等人还在文献[16]中进一步证明了矩阵秩函数可以被其凸包核范数所取代, 继而可以将问题(1)松弛为凸优化问题(2):

$$\min_{\mathbf{X} \in \mathbf{R}^{n_1 \times n_2}} \|\mathbf{X}\|_* \quad \text{s. t.} \quad P_\Omega(\mathbf{R}) = P_\Omega(\mathbf{X}) \quad (2)$$

目前对于问题(2)的求解算法大多是从向量空间压缩感知理论直接推广过来的矩阵核范数最小化方法, 如 Cai 等人<sup>[7]</sup>提出的 SVT 算法, 这也是和本文所提 LiBIMC 算法最相关的算法. SVT 算法和 LiBIMC 算法的基本思想都是基于线性 Bregman 算法求解. SVT 算法的实质是将求解压缩感知基追踪优化问题的向量空间线性 Bregman 迭代算法<sup>[17]</sup>直接推广到矩阵空间, 然后再将矩阵空间线性 Bregman 迭代与近邻前向后向分裂 (Proximal Forward Backward Splitting, PFBS) 算法<sup>[18]</sup>相结合, 算法简单易行, 特别适合大规模问题求解, 但是该算法对采样矩阵的容错性不是很好. 类似于 SVT 算法, 本文也采用了从向量空间推广而来的线性 Bregman 迭代算法, 但 SVT 算法是基于二元矩阵函数 Bregman 距离的直接推广, 而 LiBIMC 算法是基于二元矩阵函数 Bregman 距离的间接推广, 其二元矩阵函数的产生是由于噪声矩阵的引入而导致的. 因此, LiBIMC 算法和 SVT 算法的最大区别: 一方面在于求解的问题不同; 另一方面在于求解过程中所用的矩阵空间线性 Bregman 迭代算法也不同, LiBIMC 算法用到的是矩阵空间交替线性分裂 Bregman 迭代算法.

Ma 等人<sup>[8]</sup>提出的 FPCA 算法是求解问题(2)的另一种矩阵核范数最小化方法. FPCA 算法基于不动点思想, 直接采用算子分裂技术构造迭代算法求解矩阵 LASSO 模型:

$$\min_{\mathbf{X} \in \mathbf{R}^{n_1 \times n_2}} \left\{ \lambda \|\mathbf{X}\|_* + \frac{1}{2} \|P_\Omega(\mathbf{R}) - P_\Omega(\mathbf{X})\|_2^2 \right\} \quad (3)$$

该算法是受文献[19]中压缩感知的不动点迭代算法启发得到的. 然而矩阵 LASSO 对于相关性非常强的数据通常会出现不稳定的情况. Lin 等人则将矩阵补全问题视为矩阵恢复(或 Robust PCA)问题的特例, 即将矩阵补全问题建模为

$$\min_{\mathbf{X} \in \mathbf{R}^{n_1 \times n_2}} \|\mathbf{X}\|_* \quad \text{s. t.} \quad \mathbf{X} + \mathbf{E} = \mathbf{R}, P_\Omega(\mathbf{E}) = 0 \quad (4)$$

进而将求解矩阵恢复问题的 IALM 算法<sup>[9]</sup>推广到求解问题(4), IALM 算法本质上和 Chen 等人<sup>[10]</sup>提

出的 ADM 算法是一致的,因而两者具有相近的算法性能.文献[9-10]的研究结果均表明,在获得相近的补全精度的前提下,IALM 算法和 ADM 算法的收敛速度明显优于 SVT 算法和 FPCA 算法. OptSpace 算法<sup>[11]</sup>是一种有别于前四者的算法,它求解问题(1)的下述近似模型:

$$\min_{\mathbf{X} \in \mathbf{R}^{n_1 \times n_2}} \{ \|\mathbf{P}_\Omega(\mathbf{X}) - \mathbf{P}_\Omega(\mathbf{R})\|_F : \text{s. t. } \text{rank}(\mathbf{X}) \leq r \} \quad (5)$$

将低秩的要求作为约束条件,目标函数则为矩阵补全的误差,即在低秩性约束下,使得补全得到的矩阵元素尽可能地接近真实值,OptSpace 本质上是一种梯度下降算法,它面临的问题在于,矩阵的秩本来就是未知的,而在问题求解时却要求预先得到矩阵的秩信息,好在文献[20]中给出了估计矩阵秩的方法.最近,Hu 等人<sup>[12]</sup>认为现有的这些核范数最小化方法有一个共同的缺陷在于,它们都是通过同时最小化所有奇异值之和而获得结果矩阵的低秩性,这将导致所求解的结果矩阵虽然低秩但不能很好地逼近真实解.由于矩阵是否低秩主要取决于最小的那些奇异值之和,如果能通过迭代优化使得结果矩阵最小的那些奇异值之和趋近于零,那么即使那些大奇异值之和足够大也不会影响结果矩阵的低秩性.因此,他们将最小的那些奇异值之和定义为截断核范数  $\|\mathbf{X}\|_r$ ,并采用截断核范数来代替标准核范数  $\|\mathbf{X}\|_*$ ,从而将矩阵补全问题建模为如下优化问题:

$$\min_{\mathbf{X} \in \mathbf{R}^{n_1 \times n_2}} \|\mathbf{X}\|_r \quad \text{s. t. } \mathbf{P}_\Omega(\mathbf{R}) = \mathbf{P}_\Omega(\mathbf{X}) \quad (6)$$

同时针对该优化问题提出了 3 种不同的求解策略,我们将其统称为 MC\_TNNR 算法.文献[12]的实验结果表明,MC\_TNNR 算法很好地解决了所求解的结果矩阵不能精确地逼近目标矩阵真实解的问题,矩阵补全精度得到了很大程度的提高.但该算法的不足之处在于问题求解过程中要求预先估计矩阵的秩信息.

求解矩阵补全问题的另一类算法是基于矩阵分解的方法,该类方法的最大优点在于求解过程不需要执行奇异值分解,从而加速了算法的执行效率.典型的算法包括 Wen 等人<sup>[13]</sup>提出的 LMaFit (Low-rank Matrix Fitting) 算法,该算法主要求解问题(1)的如下近似模型:

$$\min_{\mathbf{U}, \mathbf{W}, \mathbf{Z}} \{ \|\mathbf{U}\mathbf{W} - \mathbf{Z}\|_F^2 : \text{s. t. } \mathbf{P}_\Omega(\mathbf{Z}) = \mathbf{P}_\Omega(\mathbf{R}) \} \quad (7)$$

其中  $\mathbf{U} \in \mathbf{R}^{n_1 \times k}$ ,  $\mathbf{W} \in \mathbf{R}^{k \times n_2}$ ,  $\mathbf{Z} \in \mathbf{R}^{n_1 \times n_2}$ , 且  $k$  是预测的秩界,如果能预先获取合适的  $k$  值,LMaFit 算法就能在较小的时间复杂度内获得相当精度的解.但是

LMaFit 算法除了需要预先估计秩界以外,还存在一个不足的地方是所求出的  $(\mathbf{U}, \mathbf{W})$  解不唯一.为了克服这个不足,Boumal 等人在文献[14]中提出了一种改进的基于 Grassmann 流形的 RTRMC 算法.文献[14]的数值实验表明,RTRMC 算法对于行列不相等的长方形矩阵补全问题尤其有效,其补全性能超过了大多数现有的矩阵补全算法.

### 3 数学预备知识

**定义 1.** 矩阵范数<sup>[21]</sup>. 假设秩为  $r$  的矩阵  $\mathbf{X} \in \mathbf{R}^{n_1 \times n_2}$  的奇异值被分解为  $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ , 其中  $\mathbf{U} = [u_1, u_2, \dots, u_r] \in \mathbf{R}^{n_1 \times r}$ ,  $\mathbf{V} = [v_1, v_2, \dots, v_r] \in \mathbf{R}^{n_2 \times r}$ ,  $\mathbf{U}^T\mathbf{U} = \mathbf{I}$ ,  $\mathbf{V}^T\mathbf{V} = \mathbf{I}$ ;  $\mathbf{\Sigma} = \text{diag}(\{\sigma_i \mid 1 \leq i \leq r\})$  且  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$ , 则:

矩阵  $\mathbf{X}$  的核范数定义为

$$\|\mathbf{X}\|_* = \sum_{i=1}^r |\sigma_i|.$$

矩阵  $\mathbf{X}$  的 Frobenius 范数定义为

$$\|\mathbf{X}\|_F = \sqrt{\sum_{i=1}^{n_1} \sum_{j=1}^{n_2} |X_{ij}|^2}.$$

矩阵  $\mathbf{X}$  的  $L_{2,1}$  范数定义为

$$\|\mathbf{X}\|_{2,1} = \sum_{i=1}^{n_1} \left( \sum_{j=1}^{n_2} X_{ij}^2 \right)^{1/2}.$$

**定义 2.** 次梯度与次微分<sup>[21]</sup>. 令  $\Psi(\mathbf{X})$  是  $\mathbf{R}^{n_1 \times n_2}$  上的实值凸函数,如果  $\mathbf{Z} \in \mathbf{R}^{n_1 \times n_2}$ , 且满足

$$\Psi(\mathbf{X}) \geq \Psi(\mathbf{X}_0) + \langle \mathbf{Z}, \mathbf{X} - \mathbf{X}_0 \rangle, \quad \forall \mathbf{X} \in \mathbf{R}^{n_1 \times n_2},$$

则称  $\mathbf{Z}$  为矩阵函数  $\Psi(\mathbf{X})$  在  $\mathbf{X}_0$  处的次梯度,所有在  $\mathbf{X}_0$  处的次梯度的集合记为  $\partial\Psi(\mathbf{X}_0)$ , 同时称  $\partial\Psi(\mathbf{X}_0)$  为  $\Psi(\mathbf{X})$  在  $\mathbf{X}_0$  处的次微分.

**定义 3.** 奇异值阈值算子<sup>[7]</sup>. 假设秩为  $r$  的矩阵  $\mathbf{X} \in \mathbf{R}^{n_1 \times n_2}$  的奇异值分解为  $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ , 如果  $\tau \geq 0$ , 则  $\tau$  对应的奇异值阈值算子定义为

$$D_\tau(\mathbf{X}) = \mathbf{U}\mathbf{S}_\tau(\mathbf{\Sigma})\mathbf{V}^T,$$

其中  $\mathbf{S}_\tau(\mathbf{\Sigma}) = \text{diag}(\{\max(0, \sigma_i - \tau)\})$ .

**定义 4.** 矩阵函数 Bregman 距离<sup>[22]</sup>. 凸函数  $J(\mathbf{X}) : \mathbf{R}^{n_1 \times n_2} \rightarrow \mathbf{R}$  在  $\mathbf{X}, \mathbf{X}^k$  两点之间的 Bregman 距离定义为

$$D_J^k(\mathbf{X}, \mathbf{X}^k) = J(\mathbf{X}) - J(\mathbf{X}^k) - \langle \mathbf{P}^k, \mathbf{X} - \mathbf{X}^k \rangle,$$

其中矩阵  $\mathbf{P}^k \in \mathbf{R}^{n_1 \times n_2}$  为次微分  $\partial J(\mathbf{X}^k)$  中的一个次梯度,  $\langle \cdot, \cdot \rangle$  为矩阵内积运算.

**定义 5.** 矩阵空间近邻算子<sup>[18]</sup>. 令  $\Psi(\mathbf{X})$  是  $\mathbf{R}^{n_1 \times n_2}$  上的实值凸函数,对任意  $\mathbf{V} \in \mathbf{R}^{n_1 \times n_2}$ ,  $\Psi(\mathbf{X})$  的

近邻算子定义为

$$\text{prox}_{\Psi(\mathbf{X})}(\mathbf{V}) = \arg \min_{\mathbf{X} \in \mathbf{R}^{n_1 \times n_2}} \left\{ \frac{1}{2} \|\mathbf{X} - \mathbf{V}\|_{\text{F}}^2 + \Psi(\mathbf{X}) \right\}.$$

**定理 1**<sup>[7]</sup>. 对任意的  $\tau > 0$ ,  $\mathbf{X}, \mathbf{Z} \in \mathbf{R}^{n_1 \times n_2}$ , 则  $\text{prox}_{\tau \|\cdot\|_*}(\mathbf{Z}) = D_{\tau}(\mathbf{Z})$ .

**定理 2**<sup>[18]</sup>. 假设  $F_1, F_2$  是矩阵空间  $\mathbf{R}^{n_1 \times n_2}$  中两个下半连续的凸函数,  $F_2$  可微且对某个  $\beta \in (0, +\infty)$  满足  $\|\nabla F_2(\mathbf{U}) - \nabla F_2(\mathbf{V})\|_{\text{F}} \leq \beta \|\mathbf{U} - \mathbf{V}\|_{\text{F}}$ , 则对于凸优化问题:

$$\min_{\mathbf{X} \in \mathbf{R}^{n_1 \times n_2}} F_1(\mathbf{X}) + F_2(\mathbf{X}) \quad (8)$$

有如下性质:

(1) 如果  $\lim_{\|\mathbf{X}\|_{\text{F}} \rightarrow +\infty} F_1(\mathbf{X}) + F_2(\mathbf{X}) = +\infty$ , 则问题(8)至少存在一个解;

(2) 如果  $F_1 + F_2$  是严格凸的, 则问题(8)至多存在一个解;

(3) 如果  $F_1, F_2$  满足条件(1)和(2), 则问题(8)存在唯一解, 且对任意的初始值  $\mathbf{X}^0$  及  $0 < \delta < 2/\beta$ , 用如下方法生成的迭代序列  $\mathbf{X}^{k+1}$  收敛到问题(8)的唯一解:

$$\mathbf{X}^{k+1} = \text{prox}_{\delta F_1(\mathbf{X})}(\mathbf{X}^k - \delta \nabla F_2(\mathbf{X}^k)),$$

其中, 性质(3)所描述的求解算法称之为近邻前向后向分裂算法(Proximal Forward Backward Splitting, PFBS).

**定理 3**<sup>[23-24]</sup>. 对任意的  $\tau > 0$ ,  $\mathbf{X}, \mathbf{X}^*, \mathbf{W} \in \mathbf{R}^{n_1 \times n_2}$ , 如果令

$$(\mathbf{X}^*)^{(i)} = \max\{\|\mathbf{W}^{(i)}\|_2 - \tau, 0\} \cdot \frac{\mathbf{W}^{(i)}}{\|\mathbf{W}^{(i)}\|_2},$$

$$i = 1, 2, \dots, n_1,$$

则  $\text{prox}_{\tau \|\cdot\|_{2,1}}(\mathbf{W}) = \mathbf{X}^*$ , 其中  $\mathbf{X}^{(i)}$  表示矩阵  $\mathbf{X}$  的第  $i$  行,  $\|\mathbf{W}^{(i)}\|_2$  表示向量  $\mathbf{W}^{(i)}$  的  $L_2$  范数.

## 4 向量空间线性 Bregman 迭代算法

Bregman 迭代起源于凸分析中一种寻求等式约束下目标函数极值的优化方法<sup>[22]</sup>. Osher 等人<sup>[25]</sup>于 2005 年首先将该方法应用于全变分图像复原, 随后被拓展应用到压缩感知<sup>[26]</sup>、图像去噪<sup>[17]</sup>等领域. 该方法已成为求解  $L_1$  范数优化问题的最有效方法之一.

假设  $J(\mathbf{u})$  和  $H(\mathbf{u})$  为  $\mathbf{R}^n$  上的两个凸函数, 其中函数  $H(\mathbf{u})$  可微,  $J(\mathbf{u})$  不可微, 考虑一般无约束优化问题:

$$\min_{\mathbf{u} \in \mathbf{R}^n} J(\mathbf{u}) + \mu H(\mathbf{u}) \quad (9)$$

根据文献[22]的策略, 可以对不可微函数  $J(\mathbf{u})$  引入向量空间 Bregman 距离, 通过如下方式迭代求

解问题(9):

$$\begin{cases} \mathbf{u}^{k+1} = \arg \min_{\mathbf{u} \in \mathbf{R}^n} D_J^{\mathbf{p}^k}(\mathbf{u}, \mathbf{u}^k) + \mu H(\mathbf{u}) \\ 0 \in \partial(D_J^{\mathbf{p}^k}(\mathbf{u}, \mathbf{u}^k) + \mu H(\mathbf{u})) \Big|_{\mathbf{u}^{k+1}} \end{cases} \quad (10)$$

将  $D_J^{\mathbf{p}^k}(\mathbf{u}, \mathbf{u}^k) = J(\mathbf{u}) - J(\mathbf{u}^k) - \langle \mathbf{p}^k, \mathbf{u} - \mathbf{u}^k \rangle$  代入式(10)后得

$$\begin{cases} \mathbf{u}^{k+1} = \arg \min_{\mathbf{u} \in \mathbf{R}^n} J(\mathbf{u}) - \langle \mathbf{p}^k, \mathbf{u} \rangle + \mu H(\mathbf{u}) \\ 0 \in \partial J(\mathbf{u}^{k+1}) - \mathbf{p}^k + \mu \nabla H(\mathbf{u}^{k+1}) \end{cases} \quad (11)$$

但是, 由于  $J(\mathbf{u})$  不可微, 迭代序列(11)仍然无法直接求解. 于是文献[26]针对迭代序列(11)引入向量空间 PFBS 算法, 并设置 PFBS 算法的迭代次数为 1, 从而提出了一种有效的向量空间线性 Bregman 迭代算法.

具体说来, 对于子优化问题:

$$\mathbf{u}^{k+1} = \arg \min_{\mathbf{u} \in \mathbf{R}^n} J(\mathbf{u}) - \langle \mathbf{p}^k, \mathbf{u} \rangle + \mu H(\mathbf{u}) \quad (12)$$

如果令  $F_1(\mathbf{u}) = J(\mathbf{u}) - \langle \mathbf{p}^k, \mathbf{u} \rangle$ ,  $F_2(\mathbf{u}) = \mu H(\mathbf{u})$ , 则根据文献[26], 可以按下式求解  $\mathbf{u}^{k+1}$ :

$$\mathbf{u}^{k+1} = \text{prox}_{\delta F_1(\mathbf{u})}(\mathbf{u}^k - \delta \nabla F_2(\mathbf{u}^k)) \quad (13)$$

因此, 迭代序列(11)可转化为

$$\begin{cases} \mathbf{u}^{k+1} = \arg \min_{\mathbf{u} \in \mathbf{R}^n} \delta F_1(\mathbf{u}) + \frac{1}{2} \|\mathbf{u} - (\mathbf{u}^k - \delta \nabla F_2(\mathbf{u}^k))\|^2 \\ 0 \in \partial\left(\delta F_1(\mathbf{u}) + \frac{1}{2} \|\mathbf{u} - (\mathbf{u}^k - \delta \nabla F_2(\mathbf{u}^k))\|^2\right) \Big|_{\mathbf{u}^{k+1}} \end{cases} \quad (14)$$

由于  $\mathbf{p}^{k+1} \in \partial J(\mathbf{u}^{k+1})$ , 式(14)可进一步化简为

$$\begin{cases} \mathbf{u}^{k+1} = \arg \min_{\mathbf{u} \in \mathbf{R}^n} \delta(J(\mathbf{u}) - \langle \mathbf{p}^k, \mathbf{u} \rangle) + \\ \frac{1}{2} \|\mathbf{u} - \mathbf{u}^k + \delta \mu \nabla H(\mathbf{u}^k)\|^2 \\ \mathbf{p}^{k+1} = \mathbf{p}^k - \frac{1}{\delta}(\mathbf{u}^{k+1} - \mathbf{u}^k) - \mu \nabla H(\mathbf{u}^k) \end{cases}$$

算法 1 详细描述了求解问题(9)的向量空间线性 Bregman 迭代算法实现步骤. 随后, Cai 等人在文献[17]中从理论上严格证明了该算法的收敛性, 并将其成功应用于图像去噪研究.

**算法 1.** 向量空间线性 Bregman 迭代算法.

输入:  $\delta > 0, \mathbf{u}^0 = \mathbf{p}^0 = 0$ , 最大迭代次数  $N$

输出:  $\mathbf{u}^{\text{optimal}}$

1. FOR  $k=0$  to  $N$

2.  $\mathbf{u}^{k+1} = \arg \min_{\mathbf{u} \in \mathbf{R}^n} \frac{1}{2} \|\mathbf{u} - \mathbf{u}^k + \delta \mu \nabla H(\mathbf{u}^k)\|^2 + \delta(J(\mathbf{u}) - \langle \mathbf{p}^k, \mathbf{u} \rangle)$ ;

3.  $\mathbf{p}^{k+1} = \mathbf{p}^k - \frac{1}{\delta}(\mathbf{u}^{k+1} - \mathbf{u}^k) - \mu \nabla H(\mathbf{u}^k)$ ;

4. END

5. RETURN  $\mathbf{u}^{\text{optimal}} \leftarrow \mathbf{u}^{N+1}$ .

本文将在第 5 节将向量空间线性 Bregman 迭代算法拓展到矩阵空间,用来解决矩阵空间二元函数凸优化问题,并在此基础上提出一类鲁棒的基于线性 Bregman 迭代的结构化噪声矩阵补全算法。

## 5 基于线性 Bregman 迭代的结构化噪声矩阵补全算法(LiBIMC)

正则化技术诞生于 20 世纪 60 年代,最初是由数学领域提出,用于解决不适定问题,基本思想是通过引入含有问题先验知识的非负辅助泛函正则化因子来增加解的稳定性<sup>[27]</sup>.随着 20 世纪 80 年代机器学习的兴起,正则化技术被广泛应用于分类器设计领域,并衍生出许多著名的算法<sup>[28]</sup>.这些算法大部分都能归纳为经验风险最小化正则化算法的框架,他们的不同之处在于所用的正则化惩罚因子,不同的正则化因子能产生不同性质的解,而如何选取正则化因子通常由问题的具体要求决定,如压缩感知理论为了得到稀疏解而采用向量  $L_0$  范数正则化因子.矩阵  $L_{2,1}$  范数正则化方法最早由 Ding 在文献<sup>[29]</sup>中提出,随后被应用于多任务学习<sup>[30-31]</sup>、张量分解<sup>[32]</sup>以及多类分类问题<sup>[33]</sup>等,作为一种全新的正则化技术,它是对传统的矩阵范数正则化理论的新发展.

如前所述,现有的矩阵补全理论或者没有考虑采样矩阵的含噪问题,或者只是简单地假设采样矩阵受到随机高斯噪声污染,并不能解决实际问题中遇到的行结构化噪声污染的问题.为了有效地平滑这类结构化噪声,将其带来的不利影响尽可能降到最低,本文通过引入矩阵  $L_{2,1}$  范数正则化因子到矩阵补全问题(2),对其目标函数施加相应的正则化约束,从而将结构化噪声矩阵补全问题建模为如下  $L_{2,1}$  范数正则化问题:

$$\min_{\mathbf{X}, \mathbf{Z} \in \mathbf{R}^{n_1 \times n_2}} \|\mathbf{X}\|_* + \lambda \|\mathbf{Z}\|_{2,1} \quad \text{s. t.} \quad P_\Omega(\mathbf{R}) = P_\Omega(\mathbf{X} + \mathbf{Z}) \quad (15)$$

其中  $\mathbf{Z}$  为行稀疏噪声矩阵,  $\|\mathbf{Z}\|_{2,1}$  为平滑噪声的矩阵  $L_{2,1}$  范数正则化因子,  $\lambda$  是被用来平衡结构化噪声和矩阵低秩程度的可调参数.由于矩阵范数同时满足齐次性和三角不等式,因此易知  $L_{2,1}$  范数正则化问题(15)为凸约束优化问题且存在全局最优解.但由于核范数和  $L_{2,1}$  范数都是不可微函数,因此该问题的求解仍然是一个待解决的难题.

为了有效地求解问题(15),本文将其转换为如

下无约束优化问题:

$$\min_{\mathbf{X}, \mathbf{Z} \in \mathbf{R}^{n_1 \times n_2}} \mu (\|\mathbf{X}\|_* + \lambda \|\mathbf{Z}\|_{2,1}) + \frac{1}{2} \|P_\Omega(\mathbf{R} - \mathbf{X} - \mathbf{Z})\|_F^2 \quad (16)$$

为描述方便,不妨令

$$J(\mathbf{X}, \mathbf{Z}) = \mu \|\mathbf{X}\|_* + \mu \lambda \|\mathbf{Z}\|_{2,1},$$

$$H(\mathbf{X}, \mathbf{Z}) = \frac{1}{2} \|P_\Omega(\mathbf{R} - \mathbf{X} - \mathbf{Z})\|_F^2.$$

则问题(16)等价于

$$\min_{\mathbf{X}, \mathbf{Z} \in \mathbf{R}^{n_1 \times n_2}} J(\mathbf{X}, \mathbf{Z}) + H(\mathbf{X}, \mathbf{Z}) \quad (17)$$

容易看到,问题(17)和问题(9)非常相似,最大的区别在于问题(9)求解的是向量空间一元凸函数优化,而问题(17)求解的是矩阵空间二元凸函数优化.

为此,我们引入矩阵空间二元凸函数 Bregman 距离来求解问题(17).

首先定义函数  $J(\mathbf{X}, \mathbf{Z})$  在  $(\mathbf{X}^k, \mathbf{Z}^k)$  处的 Bregman 距离为

$$D_j^k(\mathbf{X}, \mathbf{Z}, \mathbf{X}^k, \mathbf{Z}^k) = J(\mathbf{X}, \mathbf{Z}) - J(\mathbf{X}^k, \mathbf{Z}^k) - \langle \mu \mathbf{P}_X^k, \mathbf{X} - \mathbf{X}^k \rangle - \langle \mu \lambda \mathbf{P}_Z^k, \mathbf{Z} - \mathbf{Z}^k \rangle \quad (18)$$

其中  $\mathbf{P}_X^k \in \partial \|\mathbf{X}^k\|_*$ ,  $\mathbf{P}_Z^k \in \partial \|\mathbf{Z}^k\|_{2,1}$ ,  $\mathbf{P}^k = (\mathbf{P}_X^k, \mathbf{P}_Z^k)$ .

因此,问题(17)可按如下方式迭代求解

$$\begin{cases} (\mathbf{X}^{k+1}, \mathbf{Z}^{k+1}) = \arg \min_{\mathbf{X}, \mathbf{Z} \in \mathbf{R}^{n_1 \times n_2}} D_j^k(\mathbf{X}, \mathbf{Z}, \mathbf{X}^k, \mathbf{Z}^k) + H(\mathbf{X}, \mathbf{Z}) \\ 0 \in \partial(D_j^k(\mathbf{X}, \mathbf{Z}, \mathbf{X}^k, \mathbf{Z}^k) + H(\mathbf{X}, \mathbf{Z}))|_{\mathbf{X}^{k+1}} \\ 0 \in \partial(D_j^k(\mathbf{X}, \mathbf{Z}, \mathbf{X}^k, \mathbf{Z}^k) + H(\mathbf{X}, \mathbf{Z}))|_{\mathbf{Z}^{k+1}} \end{cases}$$

进一步将 Goldstein 和 Osher 在文献<sup>[34]</sup>中提出的向量空间分裂 Bregman 迭代算法推广到矩阵空间,同时引入交替极小化方法求解  $(\mathbf{X}^{k+1}, \mathbf{Z}^{k+1})$ , 则问题(17)可进一步采用如算法 2 所述的矩阵空间交替线性分裂 Bregman 迭代第 1 算法求解.

**算法 2.** 矩阵空间交替线性分裂 Bregman 迭代第 1 算法.

输入:  $\mathbf{Z}^0 = 0, \mathbf{P}_X^0 = 0, \mathbf{P}_Z^0 = 0, P_\Omega(\mathbf{R})$ , 最大迭代次数  $N$   
输出:  $\mathbf{X}^{\text{optimal}}, \mathbf{Z}^{\text{optimal}}$

1. FOR  $k=0$  to  $N$

2.  $\mathbf{X}^{k+1} = \arg \min_{\mathbf{X} \in \mathbf{R}^{n_1 \times n_2}} \mu \|\mathbf{X}\|_* - \mu \langle \mathbf{P}_X^k, \mathbf{X} \rangle + \frac{1}{2} \|P_\Omega(\mathbf{R} - \mathbf{X} - \mathbf{Z}^k)\|_F^2;$

3.  $\mathbf{Z}^{k+1} = \arg \min_{\mathbf{Z} \in \mathbf{R}^{n_1 \times n_2}} \mu \lambda \|\mathbf{Z}\|_{2,1} - \mu \lambda \langle \mathbf{P}_Z^k, \mathbf{Z} \rangle + \frac{1}{2} \|P_\Omega(\mathbf{R} - \mathbf{X}^{k+1} - \mathbf{Z})\|_F^2;$

4.  $\mathbf{P}_X^{k+1} = \mathbf{P}_X^k + \frac{1}{\mu} P_\Omega(\mathbf{R} - \mathbf{X}^{k+1} - \mathbf{Z}^{k+1});$

$$5. \mathbf{P}_Z^{k+1} = \mathbf{P}_Z^k + \frac{1}{\mu\lambda} P_\Omega(\mathbf{R} - \mathbf{X}^{k+1} - \mathbf{Z}^{k+1});$$

6. END

7. RETURN  $\mathbf{X}^{\text{optimal}} \leftarrow \mathbf{X}^{N+1}$ ,  $\mathbf{Z}^{\text{optimal}} \leftarrow \mathbf{Z}^{N+1}$ .

进一步,我们还可以将算法 2 转换为如下的算法 3——矩阵空间交替线性分裂 Bregman 迭代第 2 算法,定理 4 将给出算法 2 和算法 3 的等价性证明.

**算法 3.** 矩阵空间交替线性分裂 Bregman 迭代第 2 算法.

输入:  $\mathbf{Z}^0 = \mathbf{0}$ ,  $\mathbf{B}^0 = P_\Omega(\mathbf{R})$ , 最大迭代次数  $N$

输出:  $\mathbf{X}^{\text{optimal}}$ ,  $\mathbf{Z}^{\text{optimal}}$

1. FOR  $k=0$  to  $N$

$$2. \mathbf{X}^{k+1} = \arg \min_{\mathbf{X} \in \mathbf{R}^{n_1 \times n_2}} \mu \|\mathbf{X}\|_* + \frac{1}{2} \|P_\Omega(\mathbf{X} + \mathbf{Z}^k) - \mathbf{B}^k\|_F^2;$$

$$3. \mathbf{Z}^{k+1} = \arg \min_{\mathbf{Z} \in \mathbf{R}^{n_1 \times n_2}} \mu\lambda \|\mathbf{Z}\|_{2,1} + \frac{1}{2} \|P_\Omega(\mathbf{Z} + \mathbf{X}^{k+1}) - \mathbf{B}^k\|_F^2;$$

$$4. \mathbf{B}^{k+1} = \mathbf{B}^k + P_\Omega(\mathbf{R} - \mathbf{X}^{k+1} - \mathbf{Z}^{k+1});$$

5. END

6. RETURN  $\mathbf{X}^{\text{optimal}} \leftarrow \mathbf{X}^{N+1}$ ,  $\mathbf{Z}^{\text{optimal}} \leftarrow \mathbf{Z}^{N+1}$ .

**定理 4.** 矩阵空间交替线性分裂 Bregman 迭代第 1 算法等价于矩阵空间交替线性分裂 Bregman 迭代第 2 算法,即算法 2 和算法 3 等价.

证明. 详见附录.

**定理 5.** 算法 3 的不动点是问题(15)的全局最优解.

证明. 详见附录.

**定理 6.** 问题(15)的全局最优解是算法 3 的不动点.

证明. 详见附录.

定理 4~定理 6 表明,算法 2 或者算法 3 的不动点,一定是结构化噪声矩阵补全问题(15)的解.因此,接下来我们将推导出求解算法 2 不动点的具体步骤,算法 3 的具体求解步骤也可类似给出,本文不再赘述.

不难看出,就算法 2 而言,由于函数  $\|\mathbf{X}\|_*$  和  $\|\mathbf{Z}\|_{2,1}$  的不可微性,使得步 2 和步 3 中求解  $\mathbf{X}^{k+1}$  和  $\mathbf{Z}^{k+1}$  的两个子问题无法直接求解,此时可将向量空间线性 Bregman 迭代算法推广到矩阵空间,并根据定理 1 和定理 3 分别对这两个子问题求解.

**子问题 1.** 求解算法 2 中变量  $\mathbf{X}^{k+1}$ .

解:根据定理 2 性质(3),并将 PFBS 算法的迭代次数置为 1,可知

$$\mathbf{X}^{k+1} = \text{prox}_{\mu\delta \|\cdot\|_*}^{\mu\delta(P_\Omega^k, \mathbf{X})}(\mathbf{X}^k + \delta P_\Omega(\mathbf{R} - \mathbf{X}^k - \mathbf{Z}^k)),$$

化简后得

$$\mathbf{X}^{k+1} = \arg \min_{\mathbf{X} \in \mathbf{R}^{n_1 \times n_2}} \mu\delta \|\mathbf{X}\|_* +$$

$$\frac{1}{2} \|\mathbf{X} - \mathbf{X}^k - \mu\delta \mathbf{P}_X^k - \delta P_\Omega(\mathbf{R} - \mathbf{X}^k - \mathbf{Z}^k)\|_F^2 \quad (19)$$

且此时应有

$$\mathbf{P}_X^{k+1} = \mathbf{P}_X^k - \frac{1}{\mu\delta} (\mathbf{X}^{k+1} - \mathbf{X}^k - \delta P_\Omega(\mathbf{R} - \mathbf{X}^k - \mathbf{Z}^k)).$$

由于  $\mathbf{P}_X^0 = \mathbf{0}$ ,可知

$$\mathbf{P}_X^{k+1} = -\frac{1}{\mu\delta} \mathbf{X}^{k+1} + \frac{1}{\mu} \sum_{i=0}^k P_\Omega(\mathbf{R} - \mathbf{X}^i - \mathbf{Z}^i) \quad (20)$$

进一步,如果令

$$\mathbf{V}^k = \delta \sum_{i=0}^k P_\Omega(\mathbf{R} - \mathbf{X}^i - \mathbf{Z}^i) \quad (21)$$

则显然有

$$\mathbf{V}^k = \mathbf{V}^{k-1} + \delta P_\Omega(\mathbf{R} - \mathbf{X}^k - \mathbf{Z}^k) \quad (22)$$

然后,将式(21)代入式(20)可得

$$\mu\delta \mathbf{P}_X^k + \mathbf{X}^k = \mathbf{V}^{k-1} \quad (23)$$

将式(22)、式(23)代入式(19)得

$$\mathbf{X}^{k+1} = \arg \min_{\mathbf{X} \in \mathbf{R}^{n_1 \times n_2}} \mu\delta \|\mathbf{X}\|_* + \frac{1}{2} \|\mathbf{X} - \mathbf{V}^k\|_F^2 \quad (24)$$

进一步,根据定理 1 可得式(24)的解析解为

$$\mathbf{X}^{k+1} = D_{\mu\delta}(\mathbf{V}^k).$$

因此,子问题 1 可按如下方式迭代求解

$$\begin{cases} \mathbf{V}^k = \mathbf{V}^{k-1} + \delta P_\Omega(\mathbf{R} - \mathbf{X}^k - \mathbf{Z}^k) \\ \mathbf{X}^{k+1} = D_{\mu\delta}(\mathbf{V}^k) \end{cases}.$$

**子问题 2.** 求解算法 2 中变量  $\mathbf{Z}^{k+1}$ .

解:类似于子问题 1 的求解过程,容易推导出如下结论:

$$\mathbf{Z}^{k+1} = \arg \min_{\mathbf{Z} \in \mathbf{R}^{n_1 \times n_2}} \mu\lambda\delta \|\mathbf{Z}\|_{2,1} +$$

$$\frac{1}{2} \|\mathbf{Z} - \mathbf{Z}^k - \mu\lambda\delta \mathbf{P}_Z^k - \delta P_\Omega(\mathbf{R} - \mathbf{X}^{k+1} - \mathbf{Z}^k)\|_F^2 \quad (25)$$

$$\mathbf{P}_Z^{k+1} = -\frac{1}{\mu\lambda\delta} \mathbf{Z}^{k+1} + \frac{1}{\mu\lambda} \sum_{i=0}^k P_\Omega(\mathbf{R} - \mathbf{X}^{i+1} - \mathbf{Z}^i) \quad (26)$$

进一步,如果令

$$\mathbf{U}^k = \delta \sum_{i=0}^k P_\Omega(\mathbf{R} - \mathbf{X}^{i+1} - \mathbf{Z}^i) \quad (27)$$

则显然有

$$\mathbf{U}^k = \mathbf{U}^{k-1} + \delta P_\Omega(\mathbf{R} - \mathbf{X}^{k+1} - \mathbf{Z}^k) \quad (28)$$

然后,将式(27)代入式(26)可得

$$\mu\lambda\delta \mathbf{P}_Z^k + \mathbf{Z}^k = \mathbf{U}^{k-1} \quad (29)$$

将式(28)、式(29)代入式(25)后得

$$\mathbf{Z}^{k+1} = \arg \min_{\mathbf{Z} \in \mathbf{R}^{n_1 \times n_2}} \mu\lambda\delta \|\mathbf{Z}\|_{2,1} + \frac{1}{2} \|\mathbf{Z} - \mathbf{U}^k\|_F^2 \quad (30)$$

进一步,根据定理 3 可得式(30)的解析解为

$$(\mathbf{Z}^{k+1})^{(i)} = \max\{\|(\mathbf{U}^k)^{(i)}\|_2 - \mu\lambda\delta, 0\} \cdot \frac{(\mathbf{U}^k)^{(i)}}{\|(\mathbf{U}^k)^{(i)}\|_2},$$

$$i = 1, 2, \dots, n_1.$$

因此,子问题 2 可按如下方式迭代求解

$$\begin{cases} \mathbf{U}^k = \mathbf{U}^{k-1} + \delta P_{\Omega}(\mathbf{R} - \mathbf{X}^{k+1} - \mathbf{Z}^k) \\ (\mathbf{Z}^{k+1})^{(i)} = \max\{\|(\mathbf{U}^k)^{(i)}\|_2 - \mu\lambda\delta, 0\} \cdot \\ \frac{(\mathbf{U}^k)^{(i)}}{\|(\mathbf{U}^k)^{(i)}\|_2}, i = 1, 2, \dots, n_1 \end{cases}$$

至此,我们可以将前述算法归纳起来,整理得到求解问题(15)的迭代算法,并将其命名为基于线性 Bregman 迭代的结构化噪声矩阵补全算法(Linearized Bregman Iteration Algorithm for Matrix Completion with Structural Noise, LiBIMC),详细步骤如算法 4 所述.

**算法 4.** 基于线性 Bregman 迭代的结构化噪声矩阵补全算法(LiBIMC).

输入:  $P_{\Omega}(\mathbf{R}), \lambda, \mu$  和  $\delta$ , 最大迭代次数  $N$

输出:  $\mathbf{X}^{\text{optimal}}, \mathbf{Z}^{\text{optimal}}$

1. INITIALIZE  $\mathbf{X}^0 = \mathbf{0}, \mathbf{Z}^0 = \mathbf{0}, \mathbf{V}^{-1} = \mathbf{0}, \mathbf{U}^{-1} = \mathbf{0}$ ;
2. FOR  $k=0$  to  $N$
2.  $\mathbf{V}^k = \mathbf{V}^{k-1} + \delta P_{\Omega}(\mathbf{R} - \mathbf{X}^k - \mathbf{Z}^k)$ ;
3.  $\mathbf{X}^{k+1} = D_{\mu\delta}(\mathbf{V}^k)$ ;
4.  $\mathbf{U}^k = \mathbf{U}^{k-1} + \delta P_{\Omega}(\mathbf{R} - \mathbf{X}^{k+1} - \mathbf{Z}^k)$ ;
5. FOR  $i=1$  to  $n_1$
6.  $(\mathbf{Z}^{k+1})^{(i)} = \frac{(\mathbf{U}^k)^{(i)}}{\|(\mathbf{U}^k)^{(i)}\|_2} \cdot \max\{\|(\mathbf{U}^k)^{(i)}\|_2 - \mu\lambda\delta, 0\}$ ;
7. END
8. END
9. RETURN  $\mathbf{X}^{\text{optimal}} \leftarrow \mathbf{X}^{N+1}, \mathbf{Z}^{\text{optimal}} \leftarrow \mathbf{Z}^{N+1}$ .

从算法 4 容易看出, LiBIMC 算法在迭代过程中一方面可使得  $\mathbf{X}^{k+1}$  保持较好的低秩性,另一方面可使  $\mathbf{Z}^{k+1}$  保持良好的行稀疏特性,同时还能够维持  $(\mathbf{V}^k, \mathbf{U}^k)$  的稀疏性以节省存储空间,而且每一次迭代过程只涉及到一次稀疏矩阵部分奇异值分解,而对于大型稀疏矩阵的部分奇异值分解可以使用成熟的 PROPACK 软件包,它基于 Lanczos 方法,可以计算奇异值分解中前  $k$  个奇异值和奇异向量. 这些特性确保了 LiBIMC 算法的可扩展性和执行的高效率.

## 6 数值实验及结果分析

本节主要通过数值实验来验证 LiBIMC 算法的性能. 实验中用到的采样数据按照如下方式产生:

- (1) 首先针对不同的矩阵维度  $n_1, n_2$  和秩  $r$ , 随

机生成两个  $n_1 \times r$  和  $r \times n_2$  的独立同分布高斯矩阵  $\mathbf{M}_L$  和  $\mathbf{M}_R$  (均值为 0, 方差为 1), 进而得到秩为  $r$  的原始数据矩阵  $\mathbf{M} = \mathbf{M}_L \cdot \mathbf{M}_R$ ;

(2) 随机生成一个  $n_1 \times n_2$  的独立同分布高斯随机矩阵  $\mathbf{E}$  (均值为 0, 方差为 1), 然后从  $\mathbf{E}$  中随机选取  $n_1 - m$  行元素将其全部设置为 0, 则矩阵  $\mathbf{E}$  仅包含  $m$  行非零元素, 我们将如此生成的仅包含  $m$  行非零元素的矩阵  $\mathbf{E}$  记为噪声矩阵  $\mathbf{E}_{\text{noise}}$ , 同时记结构化噪声污染程度为  $p_n = m/n_1$ ;

(3) 将噪声矩阵  $\mathbf{E}_{\text{noise}}$  添加到原始数据矩阵  $\mathbf{M}$ , 生成含噪矩阵  $\mathbf{M}'$ , 使得  $\mathbf{M}' = \mathbf{M} + \mathbf{E}_{\text{noise}}$ ;

(4) 从含噪矩阵  $\mathbf{M}'$  中均匀随机选取采样元素, 采样元素的下标索引集合用  $\Omega$  表示, 并记采样率为  $p_s = |\Omega| / (n_1 \times n_2)$ , 其中  $|\Omega|$  表示集合  $\Omega$  的基数.

实验中我们采用如下指标来衡量含噪矩阵  $\mathbf{M}'$  中噪声所在行元素的含噪率:

$$r_{\text{original}}^{\text{noise}} = \|P_{\Omega^n}(\mathbf{M}' - \mathbf{M})\|_{\text{F}} / \|P_{\Omega^n}(\mathbf{M})\|_{\text{F}},$$

其中  $\Omega^n$  表示含噪元素索引集;

同时, 我们还选取了 3 个常用指标来衡量 LiBIMC 算法的性能:

(1) 缺失元素补全精度:

$$r_{\text{comp}}^{\text{miss}} = \|P_{\Omega^c}(\bar{\mathbf{M}} - \mathbf{M})\|_{\text{F}} / \|P_{\Omega^c}(\mathbf{M})\|_{\text{F}},$$

其中  $\bar{\mathbf{M}}$  表示采用 LiBIMC 算法补全后的结果矩阵,  $\Omega^c$  表示缺失元素索引集;

(2) 含噪元素纠错精度:

$$r_{\text{correct}}^{\text{noise}} = \|P_{\Omega^n}(\bar{\mathbf{M}} - \mathbf{M})\|_{\text{F}} / \|P_{\Omega^n}(\mathbf{M})\|_{\text{F}};$$

(3) 结构化噪声行位置信息辨识精度:

$$r_{\text{recog}}^{\text{line}} = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}},$$

这里定义的  $r_{\text{recog}}^{\text{line}}$  即为信息检索领域常用的评价指标 F1-Measure, 其中:

$$\text{precision} = \frac{m_{\text{true}}}{m_{\text{all}}}, \text{recall} = \frac{m_{\text{true}}}{m},$$

$m_{\text{all}}$  表示被 LiBIMC 算法识别为噪声行的总行数, 其中噪声行识别的依据是: 如果 LiBIMC 算法输出的噪声矩阵  $\mathbf{Z}^{\text{optimal}}$  中某行至少存在一个非零元素, 则该行被识别为噪声行; 反之, 如果噪声矩阵  $\mathbf{Z}^{\text{optimal}}$  中某行全为零元素, 则该行不被识别为噪声行. 进一步, 不妨假设第  $i$  行已经被 LiBIMC 算法识别为噪声行, 如果采样矩阵中第  $i$  行确实受到了噪声污染, 则称该噪声行被 LiBIMC 算法正确识别,  $m_{\text{true}}$  表示被 LiBIMC 算法正确识别的噪声行总数.

实验设计中, 考虑到数据生成、加噪和采样过程的随机性, 每一次实验均重复进行 20 次. LiBIMC



算法的收敛条件设置为最大迭代次数 150. 本文所有的实验都是在 Intel Core i7-3520M 2.90GHz CPU, 4.00GB RAM 的环境下, 使用工具软件 MATLAB 完成的. 而且, 在 LiBIMC 算法的具体实现中, 我们采用了 PROPACK 软件包来实现部分奇异值分解.

考察算法 4, 易见影响 LiBIMC 算法性能的参数共有 8 个, 它们分别是  $(n_1, n_2, r, p_s, p_n)$  和  $(\lambda, \mu, \eta)$ . 前 5 个参数  $(n_1, n_2, r, p_s, p_n)$  不是可调参数, 在数据采样时就确定好了. 后 3 个参数  $(\lambda, \mu, \eta)$  是可调参数, 目前我们还没有展开对可调参数如何自适应选取的研究, 只能根据所需要处理问题的先验知识对可调参数进行估计. 但尽管如此, 我们还是可以对这 3 个可调参数的取值做一些非形式化的分析:

(1) 参数  $\lambda$  在 LiBIMC 算法中被用来平衡结构化噪声大小和矩阵低秩程度. 一般而言, 我们设定  $\lambda \in (0, 1)$ .  $\lambda$  实际大小取决于目标矩阵的低秩程度和结构化噪声大小的比值. 从式(15)容易看出,  $\lambda$  取值越大, 将导致所求的结果矩阵秩越大, 此时采样矩阵所能容忍的结构化噪声将越小; 反之, 如果目标矩阵的秩越小, 则采样矩阵所能容忍的结构化噪声将越大, 此时应该将  $\lambda$  设置为较小值; 如果目标矩阵的秩和结构化噪声都较大, 或者目标矩阵的秩和结构化噪声都较小, 此时可通过交叉验证的方法设置  $\lambda$  值.

(2) 参数  $\mu$  在 LiBIMC 算法中被用来平衡  $(\|\mathbf{X}\|_* + \lambda\|\mathbf{Z}\|_{2,1})$  项和  $\frac{1}{2}\|P_\Omega(\mathbf{R}-\mathbf{X}-\mathbf{Z})\|_F^2$  项的大小, 其中  $\frac{1}{2}\|P_\Omega(\mathbf{R}-\mathbf{X}-\mathbf{Z})\|_F^2$  项可看成采样矩阵中除结构化噪声外的其他随机噪声. 一般而言, 我们设定  $\mu \in (500, 2000)$ . 从式(16)容易看出, 参数  $\mu$  取值越大, 将导致所求的结果矩阵秩越小, 此时采样矩阵所能容忍的其他随机噪声将越大, 同时所能容忍的结构化噪声大小可以通过参数  $\lambda$  来进行调节, 当采样矩阵所含结构化噪声较大时, 可适当选取较小的  $\lambda$  值; 此外, 如果采样矩阵同时包含较大的结构化噪声和随机噪声, 则参数  $\mu$  的取值应该设置为较大.

(3) 参数  $\delta$  被用来控制和调节 LiBIMC 算法的收敛性. 由定理 2 可知: 如果存在常数  $\beta \in (0, +\infty)$ , 使得函数  $F_2(\mathbf{X}) = \frac{1}{2}\|P_\Omega(\mathbf{R}-\mathbf{X}-\mathbf{Z}^k)\|_F^2$  在  $\mathbf{R}^{n_1 \times n_2}$  上可微且对任意的  $\mathbf{U}, \mathbf{V} \in \mathbf{R}^{n_1 \times n_2}$ , 满足

$$\|\nabla F_2(\mathbf{U}) - \nabla F_2(\mathbf{V})\|_F \leq \beta\|\mathbf{U} - \mathbf{V}\|_F$$

则当变量  $\mathbf{X}$  取任意初始值及任意  $\delta \in (0, 2/\beta)$  时, LiBIMC 算法一定收敛. 但对于函数  $F_2(\mathbf{X}) = \frac{1}{2}\|P_\Omega(\mathbf{R}-\mathbf{X}-\mathbf{Z}^k)\|_F^2$  而言, 对任意的  $\mathbf{U}, \mathbf{V} \in \mathbf{R}^{n_1 \times n_2}$ , 显然有

$$\|P_\Omega(\mathbf{U} + \mathbf{Z}^k - \mathbf{R}) - P_\Omega(\mathbf{V} + \mathbf{Z}^k - \mathbf{R})\|_F \leq 1 \cdot \|\mathbf{U} - \mathbf{V}\|_F$$

成立. 因此, 至少可以令  $\beta = 1$ , 从而设定参数  $\delta \in (0, 2)$ . 此外, 从算法 4 中第 3 步我们还可以看出, 在  $(0, 2)$  范围内, 我们可以通过调节参数  $\delta$  的大小来调节结果矩阵的秩, 一般说来, 参数  $\delta$  越大, 所求的结果矩阵秩越小, 同时算法收敛速度也越快.

接下来我们设计了 4 组不同的实验: 第 1 组实验重点考察 LiBIMC 算法对缺失元素的补全精度、对含噪元素的纠错精度以及对结构化噪声行位置信息的辨识精度; 第 2 组实验主要比较了移除结构化噪声行前后 LiBIMC 算法和 IALM、SVT、FPCA 算法对缺失元素的补全精度; 第 3 组实验重点考察了采样率的变化对 LiBIMC 算法性能所带来的影响; 第 4 组实验考察了结构化噪声污染程度对 LiBIMC 算法性能的影响.

**实验 1.** LiBIMC 算法对缺失元素的补全精度、对含噪元素的纠错精度以及对结构化噪声行位置信息的辨识性能.

实验 1 中, 为了考察 LiBIMC 算法在采样矩阵包含结构化噪声情况下对缺失元素、含噪元素的补全性能以及对噪声行位置信息的辨识能力, 我们针对  $(n_1, n_2, r, p_s, p_n, \lambda, \mu, \delta)$  设计了 5 个不同的实验, 实验结果见表 1. 从表 1 容易看出, LiBIMC 算法不仅能以足够高的精度补全缺失元素, 而且能在很大程度上对含噪元素进行纠错, 更为重要的是, LiBIMC 算法对结构化噪声行位置信息的辨识精度 (F1-Measure) 达到了 100%. 有了这些精确的噪声行位置信息, 我们就可以在移除噪声行的情形下对矩阵不含噪部分的缺失元素进行补全, 从而使得现有的矩阵补全算法都能够在结构化噪声采样的情形下得以应用. 此外, 表 1 还反映了每次实验所耗费的算法运行时间以及所求解的结果矩阵的秩, 从表 1 中可以看出, 一方面在核范数的作用下, 结果矩阵的低秩性得到了很好的保持, 另一方面从算法实时性要求来看, 算法运行效率还有待进一步提高, 我们可以在未来研究中考虑引入标准矩阵补全问题中常用的快速求解算法如 IALM 或 ADM 算法来改进结构化噪声矩阵补全问题的求解效率.

表 1 LiBIMC 算法缺失元素和含噪元素补全精度以及结构化噪声行位置信息识别性能

$n_1$	$n_2$	$r$	$p_s$	$p_n$	$\mu$	$\lambda$	$\delta$	$r_{\text{noise original}}$	$r_{\text{noise correct}}$	$r_{\text{miss comp}}$	$r_{\text{line recog}}$	结果矩阵的秩	算法运行时间/s
300	400	5	0.45	0.25	1200	0.8	1.2	4.48e-1	6.52e-2	4.91e-2	1	5	6.51
500	300	5	0.35	0.15	1200	0.7	1.2	4.47e-1	5.13e-2	3.71e-2	1	5	7.12
500	500	10	0.45	0.25	1500	0.7	1.2	3.17e-1	5.86e-2	4.13e-2	1	10	11.09
1000	1000	15	0.30	0.30	1600	0.7	1.2	2.59e-1	6.31e-2	4.92e-2	1	15	37.84
1500	1000	10	0.30	0.10	1200	0.7	1.2	3.16e-1	3.70e-2	1.66e-2	1	10	52.56

实验 2. 移除结构化噪声行前后 LiBIMC、IALM、SVT、FPCA 算法对缺失元素的补全精度比较。

实验 1 结果表明, LiBIMC 算法能精确辨识出结构化噪声所在行的位置信息, 因此我们在实验 2 中设计了 5 个实验来考察移除结构化噪声行前后 LiBIMC 算法与 IALM、SVT 和 FPCA 算法对缺失元素的补全性能, 其中, IALM 算法的实现代码来源于 [http://perception.csl.illinois.edu/matrix-rank/sample\\_code.html](http://perception.csl.illinois.edu/matrix-rank/sample_code.html), SVT 算法的实现代码来源于 <http://svt.stanford.edu/code.html>, FPCA 算法的

实现代码来源于 <http://www1.se.cuhk.edu.hk/~sqma/FPCA.html>. IALM、SVT 和 FPCA 算法的参数均按照算法设计者设定的方法选取. 实验结果见表 2. 从表 2 可以看出: (1) 移除结构化噪声前, LiBIMC 算法对缺失元素的补全精度明显优于 IALM、SVT 和 FPCA 算法将近一个数量级; (2) 移除结构化噪声行后, 采样矩阵变成了无噪声采样, 此时无论是 LiBIMC 算法还是 IALM、SVT、FPCA 算法, 对缺失元素的补全精度均得到了大幅度的提高.

表 2 IALM、SVT、FPCA 与 LiBIMC 算法移除结构化噪声行前后对缺失元素的补全精度比较

$n_1$	$n_2$	$r$	$p_s$	$p_n$	$\mu$	$\lambda$	$\delta$	IALM		SVT		FPCA		LiBIMC	
								移除噪声行前	移除噪声行后	移除噪声行前	移除噪声行后	移除噪声行前	移除噪声行后	移除噪声行前	移除噪声行后
300	400	5	0.45	0.25	1200	0.8	1.2	3.06e-1	8.75e-6	3.58e-1	1.62e-5	5.66e-1	2.97e-5	4.91e-2	3.01e-4
500	300	5	0.35	0.15	1200	0.7	1.2	2.61e-1	8.16e-6	3.17e-1	1.57e-5	5.44e-1	2.08e-5	3.71e-2	5.09e-4
500	500	10	0.45	0.25	1500	0.7	1.2	2.84e-1	7.57e-6	3.32e-1	1.56e-5	4.19e-1	2.31e-5	4.13e-2	1.36e-4
1000	1000	15	0.30	0.30	1600	0.7	1.2	4.53e-1	7.62e-6	4.98e-1	9.85e-6	4.58e-1	1.02e-5	4.92e-2	1.01e-4
1500	1000	10	0.30	0.10	1200	0.7	1.2	2.95e-1	6.33e-6	3.04e-1	2.31e-5	3.23e-1	3.28e-5	1.66e-2	6.72e-5

实验 3. 采样率对 LiBIMC 算法性能的影响。

Candes 和 Recht<sup>[6]</sup> 的研究结果表明, 对不含噪声采样的标准矩阵补全问题而言, 如果目标矩阵  $\mathbf{X}$  的阶为  $n_1 \times n_2$ , 秩为  $r$ , 其自由度为  $d_f = (n_1 + n_2 - r)r$ , 当采样数目  $m < d_f$  时, 可能存在无穷多个秩为  $r$  的矩阵满足约束条件  $P_\Omega(\mathbf{R}) = P_\Omega(\mathbf{X})$ , 这就意味着, 当采样数目低于矩阵自由度时, 即使在不含噪声情形下我们也不可能精确恢复出目标矩阵. 因此, 为了有效地考察采样率对结构化噪声矩阵补全性能的影响, 实验中我们仅考虑  $m > d_f$  的情形, 即采样元素个数必须不少于矩阵自由度. 此外, 鉴于实验 1 和实验 2 的结果已经表明 LiBIMC 算法的优势主要体现在结构化噪声行位置信息的辨识和移除结构化噪声行后对缺失元素的补全能力, 因此实验 3 主要考察 LiBIMC 算法在不同采样率下所表现出来的结构化噪声行位置信息辨识能力和移除结构化噪声行后对缺失元素的补全能力. 其中, 结构化噪声行位置信息辨识能力的考察综合采用 F1-Measure 和查全率 (Recall)、查准率 (Precision) 作为评价指标, 移除结构化噪声行后对缺失元素补全能力的考察采用缺失

元素补全精度作为评价指标. 图 1 和图 2 分别从 F1-Measure 和 Recall、Precision 的 3 个评价指标反映了秩为 5、10 和 15 的  $400 \times 400$  矩阵在结构化噪声污染率为 20% 的情形下不同的采样率对结构化噪声行位置信息辨识能力的影响, 图 3 反映了同样情形下采样率对移除结构化噪声行后缺失元素补全精度的影响. 考虑到 LiBIMC 算法参数的选取和采样率大小无关, 因此本实验中的参数设置为定值, 即  $\lambda = 0.7, \mu = 1200, \eta = 1.2$ .

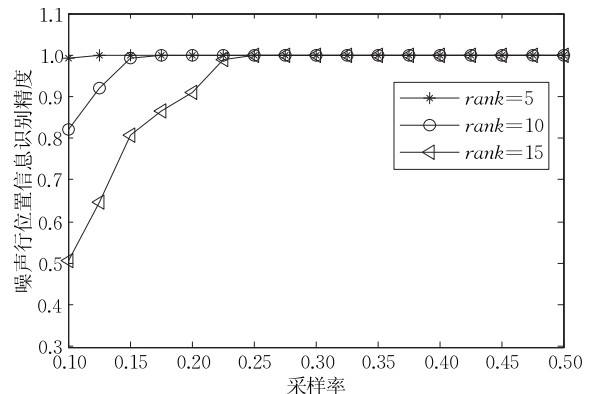


图 1 不同的采样率对结构化噪声行位置信息辨识能力 (F1-Measure) 的影响

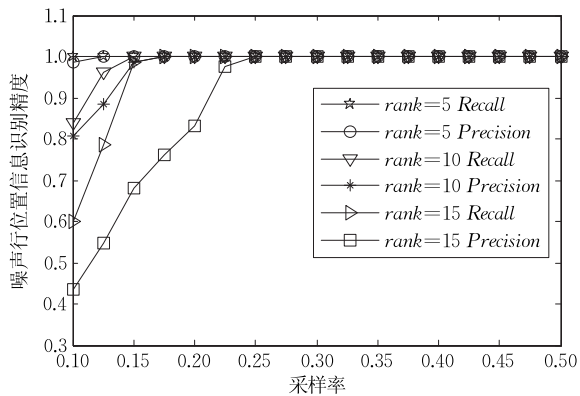


图 2 不同的采样率对结构化噪声行位置信息辨识能力(Recall & Precision)的影响

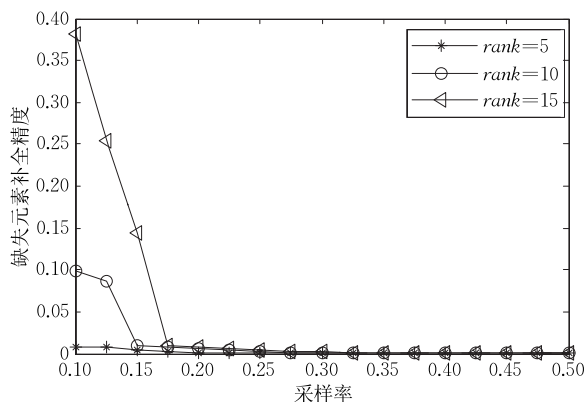


图 3 不同的采样率对移除结构化噪声行后缺失元素补全精度的影响

从图 1 可以看出:(1)当矩阵的秩较低时(例如秩为 5 时),即使采样率低至 10%,LiBIMC 算法对结构化噪声行位置信息辨识的 F1-Measure 值也能接近于 100%,这表明 LiBIMC 算法对低秩矩阵具有很强的结构化噪声行位置信息辨识能力;(2)当矩阵的秩分别提高为 10 和 15 时,如果采样率过低(例如分别低于 12.5%和 20%时),则相应的噪声行辨识能力急剧下降,但是如果将采样率各提高 5 个百分点,即分别提高到 17.5%和 25%以上,则相应的 F1-Measure 值也能达到 100%,这表明虽然随着矩阵秩的增加,在采样率保持不变的情形下, LiBIMC 算法对结构化噪声行位置信息的辨识能力会相应减弱,但实际上仍然可以通过提高采样率来减小这种负面影响,从而使得结构化噪声行位置信息得到精确辨识.进一步,从图 3 可以看出:(1)当矩阵的秩较低时(例如秩为 5 时),即使采样率低至 10%,LiBIMC 算法对缺失元素的补全能力依然很强,这表明 LiBIMC 算法对低秩含噪矩阵具有很强的缺失元素补全能力;(2)当矩阵的秩分别提高到 10 和 15 时,如果采样率较低(例如采样率为 10%和

12.5%时),LiBIMC 算法对缺失元素的补全能力急速下降.同时,我们结合图 1 和图 3 容易发现:对于矩阵秩为 15 的情形而言,当采样率分别为 17.5%和 20%时,虽然图 1 表明 LiBIMC 算法对结构化噪声行位置信息辨识的 F1-Measure 值并不高,但图 3 却反映出此时的缺失元素补全精度相当之高.究其原因,主要还是由于此时 LiBIMC 算法对结构化噪声行位置信息的查全率达到了 100%,因此图 3 中此时对移除结构化噪声行后的缺失元素补全就等同于无噪声采样情形下缺失元素的补全,自然容易获得很高的补全精度.实际上,图 2 的实验结果也证实了这一事实:无论矩阵的秩为 5,10 还是 15,在各种采样率下, LiBIMC 算法对结构化噪声行位置信息辨识的查全率都高于查准率;特别地,当秩为 15 时,如果以采样率 17.5%进行采样,此时的 F1-Measure 值和查准率均远低于 100%,但查全率仍然达到 100%,这表明在进行缺失元素补全时我们可以以较低的采样率首先移除全部噪声行(付出的相应代价是有少部分不含噪元素也会当作结构化噪声行被移除),从而得以在干净采样的情形下补全不含噪部分的缺失元素.

**实验 4.** 结构化噪声污染程度对 LiBIMC 算法性能的影响.

实验 3 考察了采样率对 LiBIMC 算法性能的影响,实验 4 将考察结构化噪声污染程度对 LiBIMC 算法的性能影响.鉴于前文对 LiBIMC 算法可调参数的分析已经表明参数  $\lambda$  的选取和结构化噪声污染程度的大小密切相关,因此本实验中仅固定参数  $\mu=1200, \eta=1.2$ ,而参数  $\lambda$  的取值是从 0.75 依噪声比例均匀递减到 0.51.图 4 反映了秩分别为 5、10 和 15 的  $400 \times 400$  矩阵在采样率为 30%和 40%情形下不同结构化噪声污染程度对噪声行位置信息辨识能力(F1-Measure)的比较.图 5 反映了秩分别为

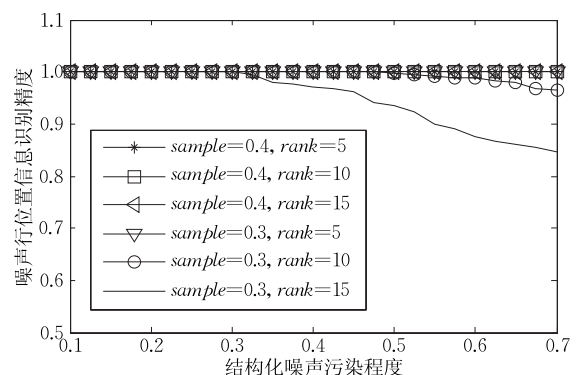


图 4 不同的噪声污染程度对结构化噪声行位置信息辨识能力(F1-Measure)的影响

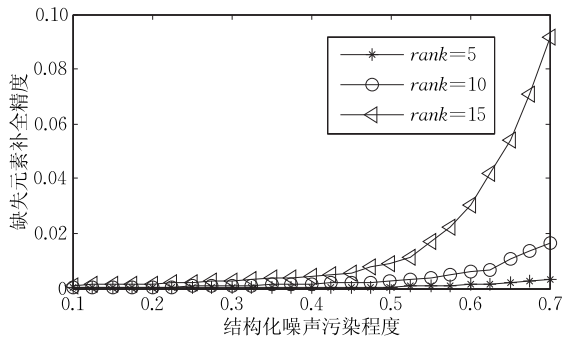


图5 采样率为40%时不同的噪声污染程度对移除结构化噪声行后缺失元素补全精度的影响

5、10和15的 $400 \times 400$ 矩阵在采样率为40%的情形下不同结构化噪声污染程度对移除结构化噪声行后缺失元素补全精度的比较。

从图4可以看出:当采样率达到40%时,对秩为5、10和15的矩阵而言,即使结构化噪声程度达到70%,LiBIMC算法对噪声行位置信息辨识的F1-Measure值仍然能达到。但当采样率降到30%时,LiBIMC算法对噪声信息的辨识能力随着矩阵秩的增大和结构化噪声污染程度的提高显著下降,这表明结构化噪声污染程度对LiBIMC算法辨识噪声行位置信息能力的影响同时取决于矩阵的低秩程度和采样率的大小,如果矩阵足够低秩,而且采样率足够高,则LiBIMC算法对噪声行位置信息的辨识能力基本不受结构化噪声污染程度的影响,反之则影响较大。进一步,从图5可以看出:当矩阵的秩为15时,即使采样率高达40%,随着结构化噪声污染程度的增加,缺失元素的补全精度下降也比较明显,但另一方面图4却表明当矩阵秩为15时,如果采样率达到40%,LiBIMC算法对噪声行位置信息辨识的F1-Measure值能达到100%,这就意味着其查全率能达到100%,也就是说在补全缺失元素时完全可以在移除全部噪声行元素的无噪声采样状态下进行,理论上此时的缺失元素补全精度是不应该随着结构化噪声污染程度的增加而不断下降的。究其原因,主要是由于当噪声含量达到一定比例时,如果移除全部噪声行信息,则矩阵的低秩性会受到很大的影响。例如对 $400 \times 400$ 的矩阵而言,如果噪声含量达到50%,则移除噪声行后所得到的干净矩阵最多变为 $200 \times 400$ ,因此其低秩性发生了很大的变化,从而使得图5所示的“当秩为15时,移除全部结构化噪声行后缺失元素补全精度依然随着噪声污染程度的提高而急剧下降”的现象产生。

最后,综合上述实验1~实验4,我们可以得出

如下结论:(1)在结构化噪声采样情形下,如果仅仅需要恢复不含噪部分的缺失元素,我们可以利用LiBIMC算法预先探测出噪声行位置,然后再选择任意一种已有的矩阵补全算法,在移除噪声行信息后,就可以以非常高的精度恢复出缺失元素;(2)如果同时需要对噪声元素纠错并且恢复含噪部分的缺失元素,可以选择LiBIMC算法进行一次性补全,虽然补全精度低于无噪情形,但与IALM、SVT和FPCA算法相比,仍然有了将近一个数量级的提高;(3)LiBIMC算法的性能受到采样率和结构化噪声污染程度的影响,特别地,矩阵的秩越高,LiBIMC算法性能受到采样率和结构化噪声污染程度的影响越明显;(4)随着矩阵秩的增加和结构化噪声污染程度的提高,虽然LiBIMC算法的补全性能会受到影响,但仍然可以通过提高元素的采样率来有效减小甚至抵消这种影响,从而达到理想的矩阵补全性能。

## 7 结 论

本文针对矩阵补全理论在实际问题中经常遇到的行结构化噪声情形提出了一种可扩充性好的有效解决方案。该方案首先通过引入 $L_{2,1}$ 范数正则化技术将其建模为一类凸约束优化问题,然后为此设计了一种鲁棒的求解算法——基于线性Bregman迭代的结构化噪声矩阵补全算法(LiBIMC)。LiBIMC算法不仅能以较高的精度补全缺失元素,而且还能对含噪元素进行纠错,更为重要的是,LiBIMC算法能够精确辨识采样矩阵中噪声元素所在行的位置信息。有了这些噪声行位置信息,已有的矩阵补全算法都可以在预先移除采样矩阵噪声行的前提下,对无噪声部分的缺失元素进行精确补全。

在接下来的工作中,我们将进一步对LiBIMC算法的收敛性进行严格的理论分析,同时研究LiBIMC算法可调参数的自适应设置方法,并展开LiBIMC算法在推荐系统、无线传感网节点定位和数据收集等方面的应用研究。

## 参 考 文 献

- [1] Wang S S, Zhang Z H. Colorization by matrix completion// Proceedings of the 26th National Conference on Artificial Intelligence (AAAI 2012). Toronto, Canada, 2012: 1-7
- [2] Li K, Dai Q, Xu W, et al. Three-dimensional motion estimation via matrix completion. IEEE Transactions on

- Systems, Man, and Cybernetics-Part B: Cybernetics, 2012, 42(2): 539-552
- [3] Cheng J, Ye Q, Jiang H B, Wang D. STCDG: An efficient data gathering algorithm based on matrix completion for wireless sensor networks. *IEEE Transactions on Wireless Communications*, 2013, 12(2): 850-861
- [4] Cabral R, De la T, Costeira J, Bernardino A. Matrix completion for multi-label image classification//*Proceedings of the 25th Annual Conference on Neural Information Processing Systems (NIPS 2011)*. Granada, Spain, 2011: 190-198
- [5] Ji H, Liu C, Shen Z, Xu Y. Robust video denoising using low rank matrix completion//*Proceedings of the 23th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2010)*. San Francisco, USA, 2010: 1797-1798
- [6] Candes E J, Recht B. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 2009, 9(6): 717-772
- [7] Cai J F, Candes E J, Shen Z. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 2010, 20(4): 1956-1982
- [8] Ma S, Goldfarb D, Chen L. Fixed point and Bregman iterative methods for matrix rank minimization. *Mathematics Programming Series A*, 2011, 128(1): 321-353
- [9] Lin Z, Chen M, Ma Y. The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices. UIUC University, USA; Technical Report UILU-ENG-09-2214, 2010
- [10] Chen C, He B, Yuan X. Matrix completion via an alternating direction method. *SIAM Journal on Numerical Analysis*, 2012, 32(1): 227-245
- [11] Keshavan R, Montannari A, Oh S. Matrix completion from a few entries. *IEEE Transactions on Information Theory*, 2010, 56(6): 2980-2998
- [12] Hu Y, Zhang D, Ye J, et al. Fast and accurate matrix completion via truncated nuclear norm regularization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013, 35(9): 2117-2130
- [13] Wen Z W, Yin W T, Zhang Y. Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. Rice University, USA; Technical Report TR1007, 2010: 1-24
- [14] Boumal N, Absil P. RTRMC: A Riemannian trust region method for matrix completion//*Proceedings of the 25th Annual Conference on Neural Information Processing Systems*. Granada, Spain, 2011: 406-414
- [15] Fazel M. Matrix Rank Minimization with Applications [Ph. D. dissertation]. Stanford University, USA, 2002
- [16] Recht B, Fazel M, Parrilo P A. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, 2010, 52(3): 471-501
- [17] Cai J F, Osher S, Shen Z. Linearized Bregman iteration for frame-based image deblurring. *SIAM Journal on Imaging Sciences*, 2009, 2: 226-252
- [18] Combettes P L, Wajs V R. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling and Simulation*, 2005, 4(4): 1168-1200
- [19] Hale E T, Yin W, Zhang Y. Fixed-point continuation for L1-minimization: Methodology and convergence. *SIAM Journal on Optimization*, 2008, 19(3): 1107-1130
- [20] Keshavan R H, Oh S. A gradient descent algorithm on the grassman manifold for matrix completion. ArXiv: 0910.5260, 2009
- [21] Boyd S, Vandenberghe L. *Convex Optimization*. Cambridge: Cambridge University Press, 2009
- [22] Bregman L. The relaxation method of finding the common points of convex sets and its application to the solution of problems in convex optimization. *Computational Mathematics and Mathematical Physics*, 1967, 7(3): 200-217
- [23] Liu G, Lin Z, Yan S, et al. Robust recovery of subspace structures by low-rank representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013, 35(1): 171-184
- [24] Yang J, Yin W, Zhang Y, et al. A fast algorithm for edge-preserving variational multichannel image restoration. *SIAM Journal on Imaging Sciences*, 2009, 2(2): 569-592
- [25] Osher S, Burger M, Goldfarb M. An iterative regularization method for total variation-based image restoration. *Multiscale Modeling and Simulation*, 2005, 4(2): 460-489
- [26] Yin W T, Osher S, Goldfarb D, et al. Bregman iterative algorithms for L1-minimization with applications to compressed sensing. *SIAM Journal on Imaging Sciences*, 2008, 1(1): 143-168
- [27] Honerkamp J, Weese J. Tikhonovs regularization method for ill-posed problems. *Continuum Mechanics and Thermodynamics*, 1990, 2(1): 17-30
- [28] Chen Z, Haykin S. On different facets of regularization theory. *Neural Computation*, 2002, 14(12): 2791-2846
- [29] Ding C, Zhou D, He X, Zha H. R1-PCA: Rotational invariant L1-norm principal component analysis for robust subspace factorization//*Proceedings of the International Conference on Machine Learning (ICML 2006)*. Pittsburgh, USA, 2006: 1-8
- [30] Argyriou A, Evgeniou T, Pontil M. Multi-task feature learning //*Proceedings of the 23th Annual Conference on Neural Information Processing Systems (NIPS 2007)*. Vancouver, Canada, 2007: 41-48
- [31] Nie F P, Huang H, Cai X, Ding C. Efficient and robust feature selection via joint L21-norms minimization//*Proceedings of the 24th Annual Conference on Neural Information Processing Systems (NIPS 2010)*. Vancouver, Canada, 2010: 1-9
- [32] Huang H, Ding C. Robust tensor factorization using R1-norm//*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2008)*. Alaska, USA, 2008: 1-8



[33] Cai X, Nie F P, Huang H, Ding C. Multi-class L2, 1-norm support vector machine//Proceedings of the 11th IEEE International Conference on Data Mining(ICDM 2011). Vancouver, Canada, 2011: 91-100

[34] Goldstein T, Osher S. The split Bregman method for L1-regularized problems. SIAM Journal on Imaging Science, 2009, 2(2): 323-343

附 录.

**定理 4.** 矩阵空间交替线性分裂 Bregman 迭代第 1 算法等价于矩阵空间交替线性分裂 Bregman 迭代第 2 算法,即算法 2 和算法 3 等价.

证明. 首先证明由算法 2 的步 2、步 4 可以等价推出算法 3 的步 2、步 4:

由于  $\mathbf{P}_X^0 = 0$ , 从算法 2 的步 4 可得

$$\mathbf{P}_X^k = \frac{1}{\mu} \sum_{i=1}^k P_{\Omega}(\mathbf{R} - \mathbf{X}^i - \mathbf{Z}^i) \quad (\text{I})$$

不妨令:

$$E(\mathbf{X}) = \mu \|\mathbf{X}\|_* - \mu \langle \mathbf{P}_X^k, \mathbf{X} \rangle + \frac{1}{2} \|P_{\Omega}(\mathbf{R} - \mathbf{X} - \mathbf{Z}^k)\|_{\text{F}}^2.$$

并将式(I)代入  $E(\mathbf{X})$  并化简得

$$E(\mathbf{X}) = \mu \|\mathbf{X}\|_* + \frac{1}{2} \|P_{\Omega}(\mathbf{X} + \mathbf{Z}^k) - (P_{\Omega}(\mathbf{R}) + \sum_{i=1}^k P_{\Omega}(\mathbf{R} - \mathbf{X}^i - \mathbf{Z}^i))\|_{\text{F}}^2 + c,$$

其中  $c$  为与  $\mathbf{X}$  无关的常数.

进一步, 如果令

$$\mathbf{B}^0 = P_{\Omega}(\mathbf{R}), \mathbf{B}^k = P_{\Omega}(\mathbf{R}) + \sum_{i=1}^k P_{\Omega}(\mathbf{R} - \mathbf{X}^i - \mathbf{Z}^i),$$

则显然有

$$\mathbf{B}^{k+1} = \mathbf{B}^k + P_{\Omega}(\mathbf{R} - \mathbf{X}^{k+1} - \mathbf{Z}^{k+1}) \quad (\text{II})$$

此即为算法 3 的步 4.

再将式(II)代入  $E(\mathbf{X})$ , 可将算法 2 的步 2 转换为算法 3 的步 2:

$$\mathbf{X}^{k+1} = \arg \min_{\mathbf{X} \in \mathbf{R}^{n_1 \times n_2}} \mu \|\mathbf{X}\|_* + \frac{1}{2} \|P_{\Omega}(\mathbf{X} + \mathbf{Z}^k) - \mathbf{B}^k\|_{\text{F}}^2.$$

同理, 由算法 2 的步 3、步 5 可以等价推出算法 3 的步 3、步 4.

综上所述, 可知算法 2 等价于算法 3. 证毕.

**定理 5.** 算法 3 的不动点是问题(15)的全局最优解.

证明. 不失一般性, 假设算法 3 存在不动点  $(\hat{\mathbf{X}}, \hat{\mathbf{Z}}, \hat{\mathbf{B}})$ , 则该不动点满足:

$$\begin{cases} \hat{\mathbf{X}} = \arg \min_{\mathbf{X} \in \mathbf{R}^{n_1 \times n_2}} \mu \|\mathbf{X}\|_* + \frac{1}{2} \|P_{\Omega}(\mathbf{X} + \hat{\mathbf{Z}}) - \hat{\mathbf{B}}\|_{\text{F}}^2 \\ \hat{\mathbf{Z}} = \arg \min_{\mathbf{Z} \in \mathbf{R}^{n_1 \times n_2}} \mu \lambda \|\mathbf{Z}\|_{2,1} + \frac{1}{2} \|P_{\Omega}(\mathbf{Z} + \hat{\mathbf{X}}) - \hat{\mathbf{B}}\|_{\text{F}}^2 \\ \hat{\mathbf{B}} = \hat{\mathbf{B}} + P_{\Omega}(\mathbf{R} - \hat{\mathbf{X}} - \hat{\mathbf{Z}}) \end{cases} \quad (\text{III})$$

再假设问题(15)的全局最优解为  $(\mathbf{X}^*, \mathbf{Z}^*)$ , 则  $(\mathbf{X}^*, \mathbf{Z}^*)$  一定满足:

$$\begin{cases} (\mathbf{X}^*, \mathbf{Z}^*) = \arg \min_{\mathbf{X}, \mathbf{Z} \in \mathbf{R}^{n_1 \times n_2}} \|\mathbf{X}\|_* + \lambda \|\mathbf{Z}\|_{2,1} \\ P_{\Omega}(\mathbf{R}) = P_{\Omega}(\mathbf{X}^* + \mathbf{Z}^*) \end{cases} \quad (\text{IV})$$

根据式(III), 显然有

$$\mu \|\hat{\mathbf{X}}\|_* + \frac{1}{2} \|P_{\Omega}(\hat{\mathbf{X}} + \hat{\mathbf{Z}}) - \hat{\mathbf{B}}\|_{\text{F}}^2 \leq$$

$$\mu \|\mathbf{X}^*\|_* + \frac{1}{2} \|P_{\Omega}(\mathbf{X}^* + \hat{\mathbf{Z}}) - \hat{\mathbf{B}}\|_{\text{F}}^2 \quad (\text{V})$$

$$\mu \lambda \|\hat{\mathbf{Z}}\|_{2,1} + \frac{1}{2} \|P_{\Omega}(\hat{\mathbf{Z}} + \hat{\mathbf{X}}) - \hat{\mathbf{B}}\|_{\text{F}}^2 \leq$$

$$\mu \lambda \|\mathbf{Z}^*\|_{2,1} + \frac{1}{2} \|P_{\Omega}(\mathbf{Z}^* + \hat{\mathbf{X}}) - \hat{\mathbf{B}}\|_{\text{F}}^2 \quad (\text{VI})$$

式(V)+式(VI)得

$$\mu \|\hat{\mathbf{X}}\|_* + \mu \lambda \|\hat{\mathbf{Z}}\|_{2,1} + \|P_{\Omega}(\hat{\mathbf{Z}} + \hat{\mathbf{X}}) - \hat{\mathbf{B}}\|_{\text{F}}^2 \leq$$

$$\mu \|\mathbf{X}^*\|_* + \mu \lambda \|\mathbf{Z}^*\|_{2,1} + \frac{1}{2} \|P_{\Omega}(\mathbf{X}^* + \hat{\mathbf{Z}}) - \hat{\mathbf{B}}\|_{\text{F}}^2 +$$

$$\frac{1}{2} \|P_{\Omega}(\mathbf{Z}^* + \hat{\mathbf{X}}) - \hat{\mathbf{B}}\|_{\text{F}}^2 \quad (\text{VII})$$

由于  $P_{\Omega}(\mathbf{R} - \hat{\mathbf{X}} - \hat{\mathbf{Z}}) = 0$  且  $P_{\Omega}(\mathbf{R}) = P_{\Omega}(\mathbf{X}^* + \mathbf{Z}^*)$ , 则式(VII)可化为

$$\mu \|\hat{\mathbf{X}}\|_* + \mu \lambda \|\hat{\mathbf{Z}}\|_{2,1} + \|P_{\Omega}(\mathbf{R}) - \hat{\mathbf{B}}\|_{\text{F}}^2 \leq$$

$$\mu \|\mathbf{X}^*\|_* + \mu \lambda \|\mathbf{Z}^*\|_{2,1} +$$

$$\frac{1}{2} \|P_{\Omega}(\hat{\mathbf{Z}}) - P_{\Omega}(\mathbf{Z}^*) + P_{\Omega}(\mathbf{R}) - \hat{\mathbf{B}}\|_{\text{F}}^2 +$$

$$\frac{1}{2} \|P_{\Omega}(\mathbf{Z}^*) - P_{\Omega}(\hat{\mathbf{Z}}) + P_{\Omega}(\mathbf{R}) - \hat{\mathbf{B}}\|_{\text{F}}^2,$$

即

$$\|\hat{\mathbf{X}}\|_* + \lambda \|\hat{\mathbf{Z}}\|_{2,1} \leq$$

$$\|\mathbf{X}^*\|_* + \lambda \|\mathbf{Z}^*\|_{2,1} + \|P_{\Omega}(\mathbf{Z}^*) - P_{\Omega}(\hat{\mathbf{Z}})\|_{\text{F}}^2 / \mu.$$

由于参数  $\mu$  是任意取值的, 所以当  $P_{\Omega}(\mathbf{R} - \hat{\mathbf{X}} - \hat{\mathbf{Z}}) = 0$  时, 恒有  $\|\hat{\mathbf{X}}\|_* + \lambda \|\hat{\mathbf{Z}}\|_{2,1} \leq \|\mathbf{X}^*\|_* + \lambda \|\mathbf{Z}^*\|_{2,1}$  成立, 因为已经假设

$$(\mathbf{X}^*, \mathbf{Z}^*) = \arg \min_{\mathbf{X}, \mathbf{Z} \in \mathbf{R}^{n_1 \times n_2}} \{\|\mathbf{X}\|_* + \lambda \|\mathbf{Z}\|_{2,1}$$

$$\text{s. t. } P_{\Omega}(\mathbf{R}) = P_{\Omega}(\mathbf{X} + \mathbf{Z})\},$$

因此可知  $(\hat{\mathbf{X}}, \hat{\mathbf{Z}})$  是问题(15)的全局最优解. 证毕.

**定理 6.** 问题(15)的全局最优解是算法 3 的不动点.

证明. 不妨令问题(15)对应的 Lagrange 函数为

$$L(\mathbf{X}, \mathbf{Z}, \mathbf{W}) = \|\mathbf{X}\|_* + \lambda \|\mathbf{Z}\|_{2,1} + \langle \mathbf{W}, P_{\Omega}(\mathbf{R} - \mathbf{X} - \mathbf{Z}) \rangle.$$

不失一般性, 假设问题(15)存在全局最优解  $(\mathbf{X}^*, \mathbf{Z}^*)$ ,

由 KKT 条件可知一定存在矩阵  $\mathbf{W}^*$ , 使得

$$\begin{cases} \mathbf{P}_X^* - P_{\Omega}(\mathbf{W}^*) = 0 \\ \lambda \mathbf{P}_Z^* - P_{\Omega}(\mathbf{W}^*) = 0 \\ P_{\Omega}(\mathbf{R}) = P_{\Omega}(\mathbf{X}^* + \mathbf{Z}^*) \end{cases} \quad (\text{VIII})$$

式(VIII)等价于

$$\begin{cases} \mathbf{P}_X^* + \frac{1}{\mu} P_{\Omega}(\mathbf{X}^* + \mathbf{Z}^*) - \frac{1}{\mu} (P_{\Omega}(\mathbf{R}) + \mu P_{\Omega}(\mathbf{W}^*)) = 0 \\ \lambda \mathbf{P}_Z^* + \frac{1}{\mu} P_{\Omega}(\mathbf{X}^* + \mathbf{Z}^*) - \frac{1}{\mu} (P_{\Omega}(\mathbf{R}) + \mu P_{\Omega}(\mathbf{W}^*)) = 0 \\ P_{\Omega}(\mathbf{R}) = P_{\Omega}(\mathbf{X}^* + \mathbf{Z}^*) \end{cases} \quad (\text{IX})$$

此时令  $\mathbf{B}^* = \mu P_{\Omega}(\mathbf{W}^*) + P_{\Omega}(\mathbf{R})$ , 则式(IX)进一步等价于

$$\begin{cases} \mu \mathbf{P}_X^* + P_\Omega(\mathbf{X}^* + \mathbf{Z}^*) - \mathbf{B}^* = 0 \\ \mu \lambda \mathbf{P}_Z^* + P_\Omega(\mathbf{X}^* + \mathbf{Z}^*) - \mathbf{B}^* = 0 \\ \mathbf{B}^* = \mathbf{B}^* + P_\Omega(\mathbf{R} - \mathbf{X}^* - \mathbf{Z}^*) \end{cases} \quad (\text{X})$$

从式(X)可以看出,  $(\mathbf{X}^*, \mathbf{Z}^*, \mathbf{B}^*)$  满足算法 3 的不动点存在条件:

$$\begin{cases} \mu \mathbf{P}_X^{k+1} + P_\Omega(\mathbf{X}^{k+1} + \mathbf{Z}^k) - \mathbf{B}^k = 0 \\ \mu \lambda \mathbf{P}_Z^{k+1} + P_\Omega(\mathbf{X}^{k+1} + \mathbf{Z}^{k+1}) - \mathbf{B}^k = 0 \\ \mathbf{B}^{k+1} = \mathbf{B}^k + P_\Omega(\mathbf{R} - \mathbf{X}^{k+1} - \mathbf{Z}^{k+1}) \end{cases}$$

所以, 问题(15)的全局最优解  $(\mathbf{X}^*, \mathbf{Z}^*)$  一定是算法 3 的不动点. 证毕.



**CHEN Lei**, born in 1975, Ph. D., associate professor. His research interests include machine learning, service computing and information security.

**CHEN Zheng-Yu**, born in 1978, Ph. D. candidate, associate professor. His research interests include information security and privacy preserving.

**XIAO Fu**, born in 1980, Ph. D., associate professor. His research interests include wireless sensor network, multi-media image retrieval.

**CHEN Song-Can**, born in 1962, Ph. D., professor. His research interests include machine learning, pattern recognition and neural computing.

**YANG Geng**, born in 1961, Ph. D., professor. His research interests include network security, parallel & distributed computing, mobile computing.

## Background

In data mining community, a wide range of data is naturally organized in matrix form. However, the real-world data is imperfect and entries are often missing. In many cases, the matrix must be filled in before any subsequent work can be done. Therefore, estimating missing entries from incomplete samples of an unknown matrix is becoming a challenging problem arising in many real world applications, such as image inpainting, video denoising, data gathering for wireless sensor networks and recommender systems, etc. This optimization problem, known as matrix completion, can be made well-depended by assuming the matrix to be low rank. The resulting rank-minimization problem is NP-hard, but it has recently been shown that the rank constraint can be replaced with a nuclear norm constraint. Because this nuclear norm problem is convex and can be optimized efficiently, there has been an amount of research over the past few years to develop optimization algorithms that perform well when no noise or only Gaussian noise is added randomly to the sample data. However, when some number of the rows is arbitrarily corrupted, it is well-known that the existing algorithms can return poor results. In this paper we study this very problem

and named it as the matrix completion problem with structural noise. The authors first introduce the  $L_{2,1}$ -norm regularization factor to smooth the structural noise, and develop a robust and efficient LiBIMC algorithm for its solution. The theoretical analysis proves that the fixed point of the LiBIMC algorithm is exactly the global optimal solution of this matrix completion problem. The numerical experiments show that the LiBIMC algorithm can better recover the matrix than the existing algorithms, and more importantly, the LiBIMC algorithm can exactly detect the positions where the rows are corrupted.

This research is supported by the National Basic Research Program (973 Program) of China under Grant No. 2011CB302903, the National Natural Science Foundation of China under Grant Nos. 61272084, 61272422, 61201163 and 61373137, the Key University Natural Science Research Project of Jiangsu Province under Grant No. 11KJA520002, the Natural Science Foundation of Jiangsu Province under Grant Nos. BK2011072 and BK20130096, and the Research Fund for the Doctoral Program of High Education under Grant No. 20113223110003.