arXiv:2305.02543v1 [math.OC] 4 May 2023

一种用于低秩矩阵恢复的预条件黎曼梯度下降算法

Fengmiao Bian* Jian-Feng Cai† Rui Zhang‡

摘要。 低秩矩阵恢复问题常见于信号处理、机器学习和成像科学等多个领域。黎曼梯度下降（RGD）算法已被证明是解决这一问题的有效算法。本文提出一种用于低秩矩阵恢复的预条件黎曼梯度下降（PRGD）算法。该预条件子简单且计算效率高，其构造方法是根据第 i 行和第 j 列的范数，对梯度矩阵的（i, j）项进行加权。在受限等距性质假设下，我们建立了 PRGD 的理论恢复保证。实验结果表明，在求解矩阵补全等低秩矩阵恢复问题时，PRGD 的速度最高可比 RGD 快十倍。

关键词。 黎曼优化，预条件黎曼梯度下降，低秩矩阵流形，低秩矩阵感知，低秩矩阵补全，相位恢复

AMS主题分类。15A23 · 15A83 · 68Q25 · 90C26 · 90C53

1 引言

本文研究低秩矩阵恢复问题。设 $\boldsymbol{X} \in \mathbb{R}^{n_1 \times n_2}$ 是一个秩为 $= r$ 的未知矩阵。低秩矩阵恢复问题的目标是通过 $\boldsymbol{y} \in \mathbb{R}^m$ 获得的一组线性测量次数，恢复矩阵 $\boldsymbol{X}$ 。

$$\boldsymbol{y} = \mathcal{A}\boldsymbol{X}. \tag{1.1}$$

其中 $\mathcal{A} : \mathbb{R}^{n_1 \times n_2} \mapsto \mathbb{R}^m$ 是一个线性算子，定义为

$$\mathcal{A}\boldsymbol{Y} = \begin{bmatrix} \langle \boldsymbol{A}_1, \boldsymbol{Y} \rangle \\ \langle \boldsymbol{A}_2, \boldsymbol{Y} \rangle \\ \vdots \\ \langle \boldsymbol{A}_m, \boldsymbol{Y} \rangle \end{bmatrix}, \quad \forall\, \boldsymbol{Y} \in \mathbb{R}^{n_1 \times n_2}, \tag{1.2}$$

其中， $\boldsymbol{A}_i \in \mathbb{R}^{n_1 \times n_2}$、 $i = 1$、2、.. . 、 m为测量矩阵， $\langle \boldsymbol{A}_i, \boldsymbol{Y} \rangle = \text{trace}(\boldsymbol{A}_i^T \boldsymbol{Y})$表示两个矩阵 $\boldsymbol{A}_i$ 和 $\boldsymbol{Y}$之间的标准内积。通常，测量次数远小于矩阵维度，即 $m \ll n_1 n_2$。

低秩矩阵恢复广泛存在于应用科学和工程学的各种应用场景中，例如矩阵补全 [10], 相位恢复 [30, 42], 量子态层析 [11, 37], 和多任务学习 [2, 53]。不同的应用场景通常需要采用不同类型的测量矩阵。例如，在矩阵补全中，观测的是未知矩阵的各个元素，而测量矩阵则是矩阵空间中的自然基。在相位恢复中，测量矩阵为秩1矩阵；而在量子态层析中，测量矩阵则由泡利矩阵的张量积构造而成。

*数学系，香港科技大学，中国香港。电子邮箱：mafm- bian@ust.hk

†数学系，香港科技大学，中国香港。电子邮箱：jf- cai@ust.hk

‡TheoryLab，中央研究院，2012年实验室，华为技术有限公司，中国香港。电子邮箱：zhangrui191@huawei.com

鉴于 $m \ll n_1 n_2$，线性系统 (1.1) 是高度欠定的。直接求逆 (1.1) 通常无法得到低秩矩阵解。然而，对于一个 $n_1 \times n_2$ 秩为 r 的矩阵，其自由度数量为 $(n_1 + n_2 - r)r$ [7]，当 $n_1 n_2$ 较小时，这一数量可能远小于 r。假设未知矩阵是低秩的，则可以唯一求解 (1.1)。事实上，在对 $\mathcal{A}$相对温和的假设下，(1.1) 存在唯一的低秩解，从而能够高效地进行低秩矩阵恢复。

近年来，开发用于低秩矩阵恢复的算法引起了广泛关注。从（1.1）中寻找低秩矩阵的一种直接方法，是寻找秩最低的矩阵，即求解

$$\min_{\boldsymbol{Z} \in \mathbb{R}^{n_1 \times n_2}} \operatorname{rank}(\boldsymbol{Z}) \quad \text{subject to} \quad \mathcal{A}(\boldsymbol{Z}) = \boldsymbol{y}.$$

然而，这个秩最小化问题是非凸且 NP难的，因此在计算上难以处理[1, 16, 19]。为克服这些问题，凸松弛已成为恢复低秩矩阵的重要方法。具体而言，可在优化问题中用 $\|\boldsymbol{Z}\|_*$ 的核范数 $\boldsymbol{Z}$（即其奇异值之和）替代 $\boldsymbol{Z}$ 的秩。由此得到以下松弛后的凸优化问题

$$\min_{\boldsymbol{Z} \in \mathbb{R}^{n_1 \times n_2}} \|\boldsymbol{Z}\|_* \quad \text{subject to} \quad \mathcal{A}(\boldsymbol{Z}) = \boldsymbol{y}. \tag{1.3}$$

在 $\mathcal{A}$的各种设定下，已有研究证明，核范数最小化（1.3）的解就是真实低秩矩阵 $\boldsymbol{X}$ [5, 6, 11, 13, 15, 17, 55]。然而，核范数函数的不光滑性使显式梯度型算法收敛缓慢。因此，更倾向于使用隐式梯度型算法或近端算法。研究表明，核范数的近端算子就是奇异值阈值化（SVT）算子 [27]。因此，基于 SVT 的算法在低秩矩阵恢复中广受欢迎，包括前向-后向分裂算法 [44], 交替方向乘子法 [8, 38, 58], 近端交替线性化最小化算法 [24] 和加速近端梯度算法 [34]。其他不依赖 SVT 的算法可参见 [9, 57]。

为提高计算效率，人们开发了基于非凸优化的算法。许多非凸优化算法旨在求解以下秩最小化问题:

$$\min_{\boldsymbol{Z} \in \mathbb{R}^{n_1 \times n_2}} \frac{1}{2}\|\mathcal{A}\boldsymbol{Z} - \boldsymbol{y}\|_2^2 \quad \text{subject to} \quad \operatorname{rank}(\boldsymbol{Z}) = r. \tag{1.4}$$

一种突出的方法是迭代硬阈值（迭代硬阈值）方法 [3, 23, 32, 40], 该方法在每一步使用硬阈值算子将迭代矩阵投影到所有秩为r 的矩阵集合上。为避免大规模奇异值分解（奇异值分解）的计算，其他一些非凸方法采用矩阵分解 $\boldsymbol{Z} = \boldsymbol{L}\boldsymbol{R}^T$ 其中 $\boldsymbol{L} \in \mathbb{R}^{n_1 \times r}$, $\boldsymbol{R} \in \mathbb{R}^{n_2 \times r}$ 来参数化未知的秩为r矩阵。于是（1.4）可改写为以下优化问题:

$$\min_{\boldsymbol{L} \in \mathbb{R}^{n_1 \times r},\ \boldsymbol{R} \in \mathbb{R}^{n_2 \times r}} \|\mathcal{A}(\boldsymbol{L}\boldsymbol{R}^T) - \boldsymbol{y}\|_2^2. \tag{1.5}$$

尽管该问题是非凸的，但已有研究表明 [18, 33, 39, 41, 43, 45, 52, 54] ，在进行适当初始化（例如采用谱初始化）后，对于矩阵感知等各种低秩矩阵恢复问题，梯度下降等简单算法都能收敛到真实的低秩矩阵 [39, 45], 矩阵补全 [33, 39, 41, 43], 相位恢复 [18, 54], 以及稳健主成分分析（稳健 PCA） [52]。然而，低秩矩阵的因子分解具有冗余性且不唯一，导致未知低秩矩阵过度参数化。因此，必须对因子施加正则化项 $\boldsymbol{L}$ 和 $\boldsymbol{R}$ ，才能找到所需的因子分解 [41, 43], ，而这些方法的收敛速度高度依赖于未知低秩矩阵的条件数。为避免过度参数化并进一步提高效率，研究者采用了黎曼流形优化算法。这些算法将未知低秩矩阵表示为商黎曼流形上的一个元素 [46, 50, 51, 56]，或表示为嵌入在$\mathbb{R}^{n_1 \times n_2}$ [7, 26, 35, 36]中的黎曼流形上的一个元素。黎曼流形优化算法能够实现更高效的低秩矩阵恢复算法，并且尽管问题具有非凸性，其理论恢复保证也得到了证明 [26, 35, 36, 39, 41, 43]。

本文重点研究嵌入流形上用于低秩矩阵恢复问题的黎曼优化算法。众所周知，所有秩为 r 的矩阵构成一个嵌入于矩阵空间 $\mathbb{R}^{n_1\times n_2}$ 中的光滑流形 $\mathcal{M}_r$。因此，低秩矩阵恢复问题可以重新表述为以下优化问题。

$$\min_{\boldsymbol{Z}\in\mathcal{M}_r} \frac{1}{2}\|\mathcal{A}\boldsymbol{Z}-\boldsymbol{y}\|_2^2. \tag{1.6}$$

求解（1.6）的一种简单而高效的算法是黎曼梯度下降（RGD），它是针对嵌入式黎曼流形上的度量进行的梯度下降 $\mathcal{M}_r$。采用环境空间 $\mathbb{R}^{n_1\times n_2}$中的典范度量，RGD 已被提出并用于研究 [7, 26, 29, 35, 36, 47]低秩矩阵恢复问题。利用 $\mathcal{M}_r$ 在典范度量下的特殊结构，RGD 每次迭代的计算成本显著降低——无需进行大规模奇异值分解计算，而且每步的计算成本与基于因子分解的优化（1.5）中的梯度下降处于同一数量级。此外，理论上已证明，当以谱方法初始化时，RGD 会线性收敛至底层低秩矩阵，且收敛因子与未知矩阵的条件数无关 [26, 29, 36, 47]。所有这些特性使 RGD 成为低秩矩阵恢复中最高效的算法之一。黎曼共轭梯度算法还可以进一步提升 RGD 的性能。

RGD 在低秩矩阵恢复中的效率高度依赖于嵌入流形 $\mathcal{M}_r$所采用的度量。前文提及的所有 RGD 均使用从环境空间 $\mathbb{R}^{n_1\times n_2}$继承而来的典范度量。因此，一个自然的问题是：

我们能否在 $\mathcal{M}_r$ 上设计一种更合适的度量，从而加快 *RGD* 的收敛速度？

我们通过采用预条件技术构造数据驱动度量，对这个问题给出了肯定的回答。由此得到的算法，即采用数据驱动度量的 RGD，称为预条件 RGD（PRGD）。数值实验表明，PRGD 的效率远高于采用典范度量的 RGD。具体而言，我们的实验显示，在矩阵补全问题中，PRGD 的速度最高可达 RGD 的十倍。在理论方面，我们还证明：若采样算子 $\mathcal{A}$ 满足受限等距性质（RIP），且以零矩阵为初始点执行一步 IHT 后进行初始化，则 PRGD 线性收敛至底层低秩矩阵。此外，线性收敛因子可以是一个与问题无关的常数。

本文的主要贡献可概括如下：

- 数据驱动度量。 我们提出了一种简单、自适应且易于计算的度量，该度量由测量数据 $\boldsymbol{y}$、采样算子 $\mathcal{A}$ 和第 t 次迭代 $\boldsymbol{X}_t$ 得出。我们的度量是将 $\mathbb{R}^{n_1\times n_2}$ 的逐元素加权度量限制在 $\mathcal{M}_r$ 的切空间上构造而成的；加权矩阵的第 (i,j) 个元素由 $\boldsymbol{X}_t$ 处梯度矩阵的第 i 行和第 j 列的二范数计算得到。该加权矩阵易于计算，与采用典范度量的 RGD 相比，每次迭代的计算成本不会增加。我们在这一新度量下应用 RGD，得到一种新颖的低秩矩阵恢复算法，称为预条件黎曼梯度下降（PRGD）算法。

- 恢复保证。 我们根据感知算子 $\mathcal{A}$ 的受限等距性质（RIP）证明了 PRGD 的局部收敛性。该收敛结果保证，当以从零开始执行一步迭代硬阈值算法作为初始值时，PRGD 会线性收敛到潜在的低秩矩阵。当 $\mathcal{A}$ 中的感知矩阵 $\boldsymbol{A}_i$、 $i=1.....m$ 为随机高斯矩阵时，我们的结果表明，若测量次数 $m\sim O(nr^2)$，则以高概率满足所需的 RIP 常数。对于非凸低秩矩阵恢复算法而言，这一样本复杂度是最优的。

- 实证研究。 我们评估了 PRGD 算法在各种低秩矩阵恢复问题，尤其是矩阵补全问题上的性能。数值结果表明，PRGD 减少了迭代次数，同时不会在每次迭代中增加过多的额外计算。

因此，PRGD 比采用典范度量的 RGD 高效得多——对于矩阵补全，PRGD 算法的速度最高可达到 RGD 算法的 10 倍。我们还证明，我们的 PRGD 算法优于 NIHT 等其他算法。

本文其余部分安排如下。在第2节中，我们将详细介绍所提出的 PRGD 算法。在第3节中，我们将建立该算法的理论恢复保证。具体而言，我们将根据感知算子的受限等距性质（RIP）证明 PRGD 的局部收敛性 $\mathcal{A}$。在第 4节中，我们通过数值实验展示 PRGD 算法的效率及其相较于现有算法的优势。在第 5节中，我们总结全文并讨论一些未来的研究方向。

在深入探讨本文的主要内容之前，我们先介绍全文中使用的一些符号。我们分别用大写字母和小写字母表示矩阵和向量，并用花体字母表示算子。对于任意 $\boldsymbol{Z} \in \mathbb{R}^{n_1 \times n_2}$，我们分别用 $\sigma_{\max}(\boldsymbol{Z})$ 和 $\sigma_{\min}(\boldsymbol{Z})$ 表示其最大和最小奇异值。 $\boldsymbol{Z}$ 的条件数记为 $\operatorname{cond}(\boldsymbol{Z}) = {}^{\sigma_{\max}(\boldsymbol{Z})}\,\sigma_{\min}(\boldsymbol{Z})$。本文中，我们用 κ表示真实低秩矩阵 $\boldsymbol{X}$的条件数。我们用 $\|\boldsymbol{Z}\|_F$表示$\boldsymbol{Z}$的弗罗贝尼乌斯范数，用 $\|\boldsymbol{Z}\|$表示 $\boldsymbol{Z}$的谱范数，并用 $\boldsymbol{Z}^T$表示 $\boldsymbol{Z}$的转置。我们将矩阵范数 $\|\cdot\|_\vee$定义为

$$\|\boldsymbol{Z}\|_\vee = \max\left\{\max_i \|\boldsymbol{Z}(i,:)\|_2, \max_j \|\boldsymbol{Z}(:,j)\|_2\right\}, \tag{1.7}$$

即，矩阵所有行和列的二范数中的最大值。对于任意 $\boldsymbol{Z}$，$\boldsymbol{Y} \in \mathbb{R}^{n_1 \times n_2}$， $\boldsymbol{Z}$ 和 $\boldsymbol{Y}$ 的内积记作 $\langle \boldsymbol{Z}, \boldsymbol{Y}\rangle$。对于任意向量 $\boldsymbol{z} \in \mathbb{R}^m$，我们将该向量的 l_2 范数记作 $\|\boldsymbol{z}\|_2$。我们将恒等算子记作 $\mathcal{I}$。对于线性算子 $\mathcal{A}$：$\mathbb{R}^{n_1 \times n_2} \to \mathbb{R}^m$，我们将 $\mathcal{A}$ 的算子范数记作 $\|\mathcal{A}\|$。我们用 $\mathcal{A}^* : \mathbb{R}^m \to \mathbb{R}^{n_1 \times n_2}$ 表示 $\mathcal{A}$ 在标准度量下的伴随算子。在矩阵恢复问题中，伴随算子 $\mathcal{A}^* : \mathbb{R}^m \to \mathbb{R}^{n_1 \times n_2}$ 具有如下形式:

$$\mathcal{A}^*(\boldsymbol{p}) = \sum_{i=1}^m p_i \boldsymbol{A}_i, \quad \text{for any} \quad \boldsymbol{p} \in \mathbb{R}^m.$$

2 算法

在本节中，我们提出一种新颖的预条件黎曼梯度下降（PRGD）算法，并对一些相关算法进行回顾。

2.1 黎曼梯度下降

我们考虑 黎曼流形 $\mathcal{M}_r$ 嵌入于 $\mathbb{R}^{n_1 \times n}$中的以下一般优化问题 [2]

$$\min_{\boldsymbol{Z} \in \mathcal{M}_r} F(\boldsymbol{Z}), \tag{2.1}$$

其中 F： $\mathbb{R}^{n_1 \times n_2} \to \mathbb{R}$ 是一个目标函数。黎曼梯度下降（RGD）是一种用于求解黎曼优化问题的常用一阶算法。从初始猜测 $\boldsymbol{X}_0$出发，RGD通过以下更新规则生成一系列迭代:

$$\boldsymbol{X}_{t+1} = \mathcal{R}\left(\boldsymbol{X}_t - \alpha_t \nabla_{\mathcal{M}_r} F(\boldsymbol{X}_t)\right), \quad t = 0, 1, 2, \ldots, \tag{2.2}$$

其中， $\mathcal{R}$ 是一个回缩算子，它将 $\boldsymbol{X}_t$ 处 $\mathcal{M}_r$ 的切空间上的矩阵映射回 $\mathcal{M}_r$上的矩阵； $\alpha_t > 0$是步长; $\nabla_{\mathcal{M}_r} F(\boldsymbol{X}_t)$是 $\boldsymbol{X}_t$处 F相对于流形 $\mathcal{M}_r$的黎曼度量的梯度。每次迭代中，从 $\boldsymbol{X}_t \in \mathcal{M}_r$开始，RGD首先在 $\boldsymbol{X}_t$处 $\mathcal{M}_r$ 的切空间中执行一步标准梯度下降，然后通过回缩算子 $\mathcal{R}$将迭代矩阵从切空间回缩到 $\mathcal{M}_r$ 上。文献中已对RGD求解黎曼优化问题进行了广泛研究。

我们回顾一下，低秩矩阵恢复问题可以表述为黎曼优化问题（1.6），其目标函数为

$$F(\boldsymbol{Z}) = \frac{1}{2}\|\mathcal{A}\boldsymbol{Z} - \boldsymbol{y}\|_2^2. \tag{2.3}$$

为了将 RGD 应用于（1.6）以求解低秩矩阵恢复问题，我们需要在 $\mathcal{M}_r$ 上选择一个黎曼度量和一个回缩算子 $\mathcal{R}$。

- 典范度量。 由于嵌入流形 $\mathcal{M}_r$ 是矩阵空间 $\mathbb{R}^{n_1\times n_2}$的子集，因此自然可以将 $\mathbb{R}^{n_1\times n_2}$ 上的典范度量用作 $\mathcal{M}_r$上的度量。任意两个矩阵 $\boldsymbol{Y}, \boldsymbol{Z} \in \mathbb{R}^{n_1\times n_2}$之间的典范内积定义为$\langle \boldsymbol{Y}, \boldsymbol{Z}\rangle = \sum_{i,j} Y_{ij}Z_{ij}$，其诱导范数为弗罗贝尼乌斯范数 $\|\boldsymbol{Y}\|_F = \langle \boldsymbol{Y}, \boldsymbol{Y}\rangle^{\frac{1}{2}}$。我们将此典范内积和范数限制到$\mathcal{M}_r$ 的切空间上，从而得到 $\mathcal{M}_r$上的典范度量。令 $\mathbb{T}_{\boldsymbol{Z}}$ 表示 $\mathcal{M}_r$ 在 $\boldsymbol{Z} \in \mathcal{M}_r$处的切空间。在 $\mathcal{M}_r$的典范度量下， $\mathbb{T}_{\boldsymbol{Z}}$ 上的内积及其诱导范数与$\mathbb{R}^{n_1\times n_2}$中的完全相同。具体而言，对于任意两个矩阵 $\boldsymbol{Y}, \boldsymbol{W} \in \mathbb{T}_{\boldsymbol{Z}}$，其内积为 $\langle \boldsymbol{Y}, \boldsymbol{W}\rangle = \sum_{i,j} Y_{ij}W_{ij}$，诱导范数为弗罗贝尼乌斯范数。在典范度量下， F 在 $\mathcal{M}_r$上的梯度可轻松计算如下:

$$\nabla_{\mathcal{M}_r} F(\boldsymbol{Z}) = \mathcal{P}_{\mathbb{T}_{\boldsymbol{Z}}}\big(\nabla F(\boldsymbol{Z})\big) = \mathcal{P}_{\mathbb{T}_{\boldsymbol{Z}}}\big(\mathcal{A}^*(\mathcal{A}\boldsymbol{Z} - \boldsymbol{y})\big),$$

其中， $\mathcal{P}_{\mathbb{T}_{\boldsymbol{Z}}}$ 是 $\mathbb{R}^{n_1\times n_2}$中的正交投影算子，而 $\nabla F(\boldsymbol{Z})$是将 F 视为函数$\mathbb{R}^{n_1\times n_2} \to \mathbb{R}$时的梯度。

- 通过截断奇异值分解进行回缩。回缩算子 $\mathcal{R}$ 用于将切空间$\mathbb{T}_{\boldsymbol{Z}}$ 上的矩阵回缩至流形 $\mathcal{M}_r$，使得当 $\boldsymbol{Y} \in \mathbb{T}_{\boldsymbol{Z}}$ 接近$\boldsymbol{Z}$时， $\mathcal{R}(\boldsymbol{Y})$与 $\boldsymbol{Y}$之间的误差受到控制。回缩算子的一个常见选择是投影到 $\mathcal{M}_r$上的投影算子，也称为最佳秩-r 近似算子或截断奇异值分解算子。具体而言，对于任意具有奇异值分解 $\boldsymbol{Y} = \sum_i \sigma_i \boldsymbol{u}_i\boldsymbol{v}_i^T(\sigma_1 \geq \sigma_2 \geq \ldots)$的 $\boldsymbol{Y} \in \mathbb{T}_{\boldsymbol{Z}}$，我们选择

$$\mathcal{R}(\boldsymbol{Y}) = \mathcal{H}_r(\boldsymbol{Y}) := \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i\boldsymbol{v}_i^T.$$

请注意，回缩算子还有其他选择。

通过选择典范度量和截断奇异值分解回缩，我们可以获得如下所示的用于低秩矩阵恢复的高效RGD算法:

$$\boldsymbol{X}_{t+1} = \mathcal{H}_r\Big(\boldsymbol{X}_t - \alpha_t \mathcal{P}_{\mathbb{T}_{\boldsymbol{X}_t}}\big(\mathcal{A}^*(\mathcal{A}\boldsymbol{X}_t - \boldsymbol{y})\big)\Big), \quad t = 0, 1, 2, \ldots. \tag{2.4}$$

利用 2.4 中切空间的结构，可以非常高效地完成其计算 $\mathcal{M}_r$。除了应用 $\mathcal{A}$ 和 $\mathcal{A}^*$，式（2.4）中最耗时的操作是计算 $\mathcal{P}_{\mathbb{T}_{\boldsymbol{X}_t}}$ 和 $\mathcal{H}_r$。下面将介绍它们的计算方法。

- 投影 $\mathcal{P}_{\mathbb{T}}$的计算。 回顾一下，点 $\boldsymbol{Z} \in \mathcal{M}_r$ 处的切空间 $\mathbb{T}_{\boldsymbol{Z}}$，利用紧奇异值分解$\boldsymbol{Z} = \boldsymbol{U}_{\boldsymbol{Z}}\boldsymbol{\Sigma}_{\boldsymbol{Z}}\boldsymbol{V}_{\boldsymbol{Z}}^T$，可显式表示为

$$\mathbb{T}_{\boldsymbol{Z}} = \big\{\boldsymbol{U}_{\boldsymbol{Z}}\boldsymbol{Q}^T + \boldsymbol{P}\boldsymbol{V}_{\boldsymbol{Z}}^T \mid \boldsymbol{P} \in \mathbb{R}^{n_1\times r},\ \boldsymbol{Q} \in \mathbb{R}^{n_2\times r}\big\}. \tag{2.5}$$

因此，正交投影算子 $\mathcal{P}_{\mathbb{T}_{\boldsymbol{Z}}}$ 有如下闭式表达:

$$\mathcal{P}_{\mathbb{T}_{\boldsymbol{Z}}}\boldsymbol{Y} = \boldsymbol{U}_{\boldsymbol{Z}}\boldsymbol{U}_{\boldsymbol{Z}}^T\boldsymbol{Y} + \boldsymbol{Y}\boldsymbol{V}_{\boldsymbol{Z}}\boldsymbol{V}_{\boldsymbol{Z}}^T - \boldsymbol{U}_{\boldsymbol{Z}}\boldsymbol{U}_{\boldsymbol{Z}}^T\boldsymbol{Y}\boldsymbol{V}_{\boldsymbol{Z}}\boldsymbol{V}_{\boldsymbol{Z}}^T.$$

因此，当存在 $\boldsymbol{X}_t$ 的紧奇异值分解时，计算 $\mathcal{P}_{\mathbb{T}_{\boldsymbol{X}_t}}$ 式（2.4）只需进行 $O(r)$次矩阵-向量乘积。

- 截断奇异值分解 $\mathcal{H}_r$的计算。 注意， $\mathcal{H}_r$ 式（2.4）仅作用于矩阵 $\boldsymbol{Y} \in \mathbb{T}_{\boldsymbol{X}_t}$。由式（2.5）可知，切空间中的矩阵的秩至多为 $2r$。因此，对某些 $\boldsymbol{Y} = \boldsymbol{L}\boldsymbol{R}^T$ 和 $\boldsymbol{L} \in \mathbb{R}^{n_1 \times 2r}$，我们可以写成 $\boldsymbol{R} \in \mathbb{R}^{n_2 \times 2r}$。秩为 r 的最佳近似可以高效地按以下方式计算：$2r$ $\boldsymbol{Y}$

  – 我们计算 QR分解 $\boldsymbol{L} = \boldsymbol{Q}_1\boldsymbol{R}_1$ 和 $\boldsymbol{R} = \boldsymbol{Q}_2\boldsymbol{R}_2$。
  – 然后，$\boldsymbol{Y} = \boldsymbol{Q}_1\boldsymbol{R}_1\boldsymbol{R}_2^T\boldsymbol{Q}_2^T$。由于 $\boldsymbol{Q}_1$ 和 $\boldsymbol{Q}_2$ 正交，

$$\mathcal{H}_r(\boldsymbol{Y}) = \mathcal{H}_r(\boldsymbol{Q}_1\boldsymbol{R}_1\boldsymbol{R}_2^T\boldsymbol{Q}_2^T) = \boldsymbol{Q}_1 \cdot \mathcal{H}_r(\boldsymbol{R}_1\boldsymbol{R}_2^T) \cdot \boldsymbol{Q}_2^T.$$

  要得到 $\mathcal{H}_r(\boldsymbol{Y})$，我们只需对 $2r \times 2r$ 矩阵 $\boldsymbol{R}_1\boldsymbol{R}_2^T$进行奇异值分解。

因此， $\mathcal{H}_r$ 在（2.4）中的计算可以通过对宽度为 r 的高矩阵进行两次 QR分解、对一个 $2r \times 2r$ 矩阵进行一次奇异值分解，以及少量矩阵-向量乘积，高效完成。

因此，（2.4）每次迭代的计算成本都很低。理论分析 [26, 35, 36] 表明，在适当的 $\mathcal{A}$ 假设下，以 $\mathbf{0}$ 为初始值进行一步迭代硬阈值后初始化的 RGD（2.4）会线性收敛到真实的低秩矩阵。此外，线性收敛的收缩因子具有普适性，且独立于未知的低秩解，这意味着 RGD（2.4）只需很少的迭代次数就能获得非常精确的解。总而言之，RGD（2.4）是低秩矩阵恢复中最高效的算法之一。

RGD（2.4）及其变体已针对各种低秩矩阵恢复问题提出并得到研究，例如一般矩阵感知 [36], 矩阵补全 [7, 35], 和相位恢复 [26]。它还被扩展到其他一些相关问题，包括低秩张量恢复 [12, 28], 谱压缩感知 [29],和稳健主成分分析 [22, 48]。

2.2 预条件黎曼梯度下降

在希尔伯特空间上进行优化时，梯度下降的性能取决于底层希尔伯特空间的度量。改善迭代算法性能的一种常用技术是修改度量。在求解对称正定线性系统时，我们通常将其表述为二次优化问题 $\mathbb{R}^n$，而最速下降算法就是采用精确线搜索的梯度下降。为加速最速下降，我们改变 $\mathbb{R}^n$ 的度量，并以系数矩阵的近似加权，从而得到预条件最速下降。这种预条件技术可以与更先进的算法结合，得到最先进的线性系统求解器，例如预条件共轭梯度（PCG）法。在非线性无约束优化中，我们通常通过改变度量（例如，使用近似海森矩阵加权）来加速梯度下降算法，由此得到包括牛顿型算法在内的预条件梯度法 [4, 31]。

我们采用类似的思路，通过改变黎曼流形的度量来加速 RGD（2.4），从而得到预条件 RGD（PRGD）。由于（2.3）中的目标函数 F 是对线性系统（1.1）进行的最小二乘拟合，而 RGD 在切空间 $\mathbb{T}_{\boldsymbol{X}_t}$上对 F 进行线性化，因此收敛速度取决于感知算子 $\mathcal{A}$ 限制在 $\mathbb{T}_{\boldsymbol{X}_t}$上的条件数。设 $\mathfrak{g}$ 为$\mathcal{M}_r$ 上使用的度量（即，$\langle\cdot,\cdot\rangle_{\mathfrak{g}}$ 是 $\mathcal{M}_r$ 的切空间上的内积，$\|\cdot\|_{\mathfrak{g}}$ 是诱导范数）。那么，$\mathcal{A}$ 限制在切空间 $\mathbb{T}_{\boldsymbol{X}_t}$ 上的条件数为 $\mathrm{cond}_{\mathfrak{g},\,\mathbb{T}_{\boldsymbol{X}_t}}(\mathcal{A}) = \frac{C_{l,\mathfrak{g}}}{C_{u,\mathfrak{g}}}$，其中

$$C_{l,\mathfrak{g}}\|\boldsymbol{Z}\|_{\mathfrak{g}}^2 \le \|\mathcal{A}\boldsymbol{Z}\|_2^2 \le C_{u,\mathfrak{g}}\|\boldsymbol{Z}\|_{\mathfrak{g}}^2, \qquad \forall\ \boldsymbol{Z} \in \mathbb{T}_{\boldsymbol{X}_t}.$$

$\mathrm{cond}_{\mathfrak{g},\,\mathbb{T}_{\boldsymbol{X}_t}}(\mathcal{A})$ 越小，收敛越快。我们的目标是找到一个度量 $\mathfrak{g}$，使 $\mathrm{cond}_{\mathfrak{g},\,\mathbb{T}_{\boldsymbol{X}_t}}(\mathcal{A})$ 最小。采用典范度量（记作 $\mathfrak{g}_0$），对于许多低秩矩阵恢复问题，我们可以证明

$$(1-\delta)\|\boldsymbol{Z}\|_F^2 \le \|\mathcal{A}\boldsymbol{Z}\|_2^2 \le (1+\delta)\|\boldsymbol{Z}\|_F^2, \qquad \forall\ \boldsymbol{Z} \in \mathbb{T}_{\boldsymbol{X}_t} \tag{2.6}$$

对于某个 $\delta \in (0,1)$。在矩阵感知和量子态层析问题中，可以证明 $\mathcal{A}$满足受限等距性质（RIP），这意味着当其限制在该范围内时，它近似保持等距。

所有低秩矩阵的集合。由于任意切空间 $\mathcal{M}_r$ 上的所有矩阵的秩至多为 $2r$，因此满足 (2.6)。在矩阵补全和相位恢复中，已经证明，当 $\boldsymbol{X}_t$ 足够接近 $\boldsymbol{X}$，且 $\mathcal{A}$ 满足某些概率模型时，(2.6) 以高概率成立。不等式 (2.6) 通常是许多凸优化和非凸优化算法的恢复保证证明中的一个关键条件，这些算法包括 RGD、基于因子分解的梯度下降以及景观分析。此外，在矩阵补全、量子态层析和矩阵感知等许多应用中，只要测量次数 m 足够大，(2.6) 中的常数 δ 就可以任意小。通常，m 和 δ 的依赖关系为 $m \sim O(\delta^{-2})$。当 (2.6) 中的 δ 足够小时，$\mathcal{A}$ 在典范度量 $\mathfrak{g}_0$ 下的受限条件数为 $\mathrm{cond}_{\mathfrak{g}_0,\mathbb{T}_{\boldsymbol{X}_t}}(\mathcal{A}) = \frac{1+\delta}{1-\delta}$，这是一个较小的值。因此，当 m 足够大，使得 (2.6) 中的 δ 足够小时，在典范度量下的 RGD (2.4) 是解决低秩矩阵恢复问题最高效的算法之一。

然而，在相位恢复等其他一些应用中，即使 m 足够大，式（2.6）中的常数 δ 也不可能很小。此外，在实际场景中，我们可能没有足够多的测量次数 m，无法使式（2.6）中的 δ 足够小，从而导致受限条件数 $\mathrm{cond}_{\mathfrak{g}_0,\mathbb{T}_{\boldsymbol{X}_t}}(\mathcal{A})$ 较大。而且，式（2.6）是在关于 $\mathcal{A}$ 的特定概率模型下建立的，并以一定概率成立。实际上，我们只能获得 $\mathcal{A}$ 的某个特定实现；即使 m 不小，可用实现 $\mathcal{A}$ 仍有可能使式（2.6）在较大的 δ 下成立。在所有这些情况下，RGD（2.4）都可能收敛缓慢。为解决这一问题，我们提出采用非典范度量 $\mathfrak{g}$，使得在 $\mathfrak{g}$ 下受限条件数 $\mathrm{cond}_{\mathfrak{g},\mathbb{T}_{\boldsymbol{X}_t}}(\mathcal{A})$ 较小。这种方法有望改善 RGD 算法在式（2.6）中的常数 δ 较大且 RIP 不太满足的场景下的性能。

2.2.1 数据驱动度量构造。

我们的度量 $\mathfrak{g}$ 由测量算子 $\mathcal{A}$、测量向量 $\boldsymbol{y}$和迭代矩阵构造而成，因此是一种数据驱动度量，能够根据测量数据和算法数据进行调整。为简化记号，我们用 $\mathbb{T}_t$ 表示所有 $\mathbb{T}_{\boldsymbol{X}_t}$ 对应的切空间 $t \in \mathbb{N}$。由于只有 $\mathbb{T}_t$ 上的度量会影响 RGD 迭代（2.2），我们只需在 $\mathfrak{g}$ 上构造 $\mathbb{T}_t$。

在每个切空间 $\mathbb{T}_t$ 上构造度量 $\mathfrak{g}$ 包含两个步骤。第一步，针对线性空间优化问题 $\min_{\boldsymbol{Z}\in\mathbb{R}^{n_1\times n_2}} F(\boldsymbol{Z})$，运用预条件技术在环境空间 $\mathbb{R}^{n_1\times n_2}$ 上构造度量。第二步，将 $\mathbb{R}^{n_1\times n_2}$ 上的度量限制到切空间 $\mathbb{T}_t$上，从而得到度量 $\mathfrak{g}$。下面将详细介绍这两个步骤。

- 通过预条件在 $\mathbb{R}^{n_1\times n_2}$ 上构造度量。为构造我们的度量 $\mathfrak{g}$，我们首先运用预条件技术，为线性空间上的矩阵优化问题在环境空间 $\mathbb{R}^{n_1\times n_2}$ 上构造度量。不考虑低秩约束时，（2.1）中的优化问题就变成线性空间 $\mathbb{R}^{n_1\times n_2}$上的典型光滑优化问题:

$$\min_{\boldsymbol{Z}\in\mathbb{R}^{n_1\times n_2}} F(\boldsymbol{Z}), \qquad F(\boldsymbol{Z}) = \frac{1}{2}\|\mathcal{A}\boldsymbol{Z} - \boldsymbol{y}\|_2^2.$$

  为了解决这个问题，预条件梯度方法已发展成为一类强大的算法，包括牛顿法和拟牛顿法 [4, 31]。然而，在牛顿法和拟牛顿法中构造预条件子代价高昂。为克服这一问题，在现代机器学习（尤其是深度神经网络训练）的推动下，人们也提出了高效且有效的预条件算法。

  设 $\boldsymbol{G}_t$ 是 $\mathbb{R}^{n_1\times n_2}$ 中 F 的梯度，即，

$$\boldsymbol{G}_t = \mathcal{A}^*(\mathcal{A}(\boldsymbol{X}_t) - \boldsymbol{y}).$$

  AdaGrad [25]通过累积历史梯度的外积来构造其预条件子。它对历史梯度赋予了较高权重，从而抑制算法再次沿历史方向进行更新。这使得 AdaGrad 在实践中成为一种高效的方法。然而，计算

两个矩阵的外积仍然十分昂贵，使得 AdaGrad预条件器不适用于我们的矩阵优化。为解决这一问题，Shampoo 预条件器 [49] 被提出。Shampoo 使用张量积近似 AdaGrad预条件器。具体而言，它定义了

$$\boldsymbol{L}_t = \epsilon \boldsymbol{I}_{n_1} + \sum_{i=1}^{t} \boldsymbol{G}_i \boldsymbol{G}_i^T, \quad \text{and} \quad \boldsymbol{R}_t = \epsilon \boldsymbol{I}_{n_2} + \sum_{i=1}^{t} \boldsymbol{G}_i \boldsymbol{G}_i^T. \tag{2.7}$$

11 – – 1因此，预条件子为 $\boldsymbol{R}_t^4 \otimes \boldsymbol{L}_t^4$,- 其中在加权内积中，行和列分别由 $\boldsymbol{L}_t^4$和1$\boldsymbol{R}_t^4$ 加权。此外，行（列）还通过历史梯度的行（列）的外积累积进行预条件处理。

对于低秩矩阵恢复而言，Shampoo 的开销仍然过高，尤其是在大规模问题中。事实上，在矩阵补全中，矩阵规模如此之大，以至于普通计算机无法存储一个完整的稠密矩阵；然而，Shampoo 需要完整的稠密矩阵 $\boldsymbol{L}_t$ 和 $\boldsymbol{R}_t$。因此，为进一步降低时间和空间复杂度，我们对 Shampoo 预条件器作了进一步简化。首先，我们只使用 $\boldsymbol{X}_t$ 处的梯度，而不使用 Shampoo 中涉及的所有历史梯度。其次，我们只使用 Shampoo 中预条件矩阵的对角部分，以减少计算量和内存占用。具体而言，我们用对角近似替换 Shampoo 中的 $\boldsymbol{L}_t$ 和 $\boldsymbol{R}_t$。

$$\boldsymbol{L}_t = \epsilon \boldsymbol{I}_{n_1} + \operatorname{diag}(\boldsymbol{G}_t \boldsymbol{G}_t^T) \quad \text{and} \quad \boldsymbol{R}_t = \epsilon \boldsymbol{I}_{n_2} + \operatorname{diag}(\boldsymbol{G}_t^T \boldsymbol{G}_t). \tag{2.8}$$

1 1然后，在加权内积中，分别通过 $\boldsymbol{L}_t^4$和 $\boldsymbol{R}_t^4$对行和列进行加权，从而得到我们在 $\mathbb{R}^{n_1 \times n_2}$ 中的度量——对于任意 $\boldsymbol{Z}, \boldsymbol{Y} \in \mathbb{R}^{n_1 \times n_2}$,

$$\langle \boldsymbol{Z}, \boldsymbol{Y} \rangle_{\mathcal{W}_t} = \langle \mathcal{W}_t \boldsymbol{Z}, \boldsymbol{Y} \rangle = \langle \boldsymbol{L}_t^{\frac{1}{4}} \boldsymbol{Z} \boldsymbol{R}_t^{\frac{1}{4}}, \boldsymbol{Y} \rangle. \tag{2.9}$$

显然，我们度量中的加权算子 $\mathcal{W}_t$ 是逐元素应用的。值得注意的是，对角矩阵 $\operatorname{diag}(\boldsymbol{G}_t \boldsymbol{G}_t^T)$ 的每个对角元素都等于 $\|\boldsymbol{G}_t(i,:)\|_2^2$，其中 $\boldsymbol{G}_t(i,:)$ 表示梯度 $\boldsymbol{G}_t$ 的第 i 行。类似地，$\operatorname{diag}(\boldsymbol{G}_t^T \boldsymbol{G}_t)$ 的第 j 个对角元素是梯度 $\boldsymbol{G}_t$ 的第 j 列的范数平方。因此，我们度量中的 $\mathcal{W}_t$ 根据梯度 $\boldsymbol{G}_t$ 的第 i 行和第 j 列的二范数，对矩阵的第 (i, j) 个元素重新加权。

- 通过限制在 $\mathbb{T}_t$ 上构造度量。我们只需将先前在 $\mathbb{R}^{n_1 \times n_2}$ 中构造的度量限制到切空间 $\mathbb{T}_t$ 上。对于任意 $\boldsymbol{Z}$, $\boldsymbol{Y} \in \mathbb{T}_t$，内积可表示如下：

$$\langle \boldsymbol{Z}, \boldsymbol{Y} \rangle_{\mathcal{W}_t} = \langle \mathcal{W}_t \boldsymbol{Z}, \boldsymbol{Y} \rangle = \langle \boldsymbol{L}_t^{\frac{1}{4}} \boldsymbol{Z} \boldsymbol{R}_t^{\frac{1}{4}}, \boldsymbol{Y} \rangle, \tag{2.10}$$

其中 $\boldsymbol{L}_t$ 和 $\boldsymbol{R}_t$ 定义于（2.8）中。由内积 $\langle \cdot, \cdot \rangle_{\mathcal{W}_t}$ 诱导的范数记为$\| \cdot \|_{\mathcal{W}_t}$。

2.2.2 新度量下的黎曼梯度下降。

要使用新度量执行 RGD（2.2），我们需要计算黎曼梯度 $\nabla_{\mathcal{M}_r} F(\boldsymbol{X}_t)$ 在新度量下的表达式。为此，设 $\boldsymbol{X}_t(s)$， $s \in \mathbb{R}$ 是 $\mathcal{M}_r$ 上的一条光滑曲线，且满足 $\boldsymbol{X}_t(0) = \boldsymbol{X}_t$。设~$\mathcal{P}_{\mathbb{T}_t}$ 是关于 $\mathbb{T}_t$ 在 $\mathbb{R}^{n_1 \times n_2}$ 中、度量（2.10）下的正交投影算子。对 $F(\boldsymbol{X}_t(s))$ 关于 s 求导可得

$$\frac{d}{ds} F(\boldsymbol{X}_t(s))\Big|_{s=0} = \langle \dot{\boldsymbol{X}}_t(0), \boldsymbol{G}_t \rangle = \langle \dot{\boldsymbol{X}}_t(0), \mathcal{W}_t^{-1} \boldsymbol{G}_t \rangle_{\mathcal{W}_t} = \langle \dot{\boldsymbol{X}}_t(0), \widetilde{\mathcal{P}}_{\mathbb{T}_t} \mathcal{W}_t^{-1} \boldsymbol{G}_t \rangle_{\mathcal{W}_t},$$

其中最后一项中，$\dot{\boldsymbol{X}}_t(0)$ 和 $\widetilde{\mathcal{P}}_{\mathbb{T}_t} \mathcal{W}_t^{-1} \boldsymbol{G}_t$ 均属于 $\mathbb{T}_t$，内积则是我们在（2.10）中构造的内积。因此，新度量下的黎曼梯度为

$$\nabla_{\mathcal{M}_r} F(\boldsymbol{X}_t) = \widetilde{\mathcal{P}}_{\mathbb{T}_t} \mathcal{W}_t^{-1} \boldsymbol{G}_t = \widetilde{\mathcal{P}}_{\mathbb{T}_t} (\boldsymbol{L}_t^{-\frac{1}{4}} \boldsymbol{G}_t \boldsymbol{R}_t^{-\frac{1}{4}}).$$

我们仍将截断奇异值分解作为回缩算子。综合以上内容，我们得到针对低秩矩阵恢复所提出的预条件 RGD（PRGD）。完整的 PRGD 算法总结于算法1。

算法 1： 预条件黎曼梯度下降（PRGD）。

初始化 $\boldsymbol{X}_0 = \mathcal{H}_r(\mathcal{A}^*\boldsymbol{y})$。

当停止条件未满足**时，执行**

计算梯度： $\boldsymbol{G}_t = \mathcal{A}^*(\mathcal{A}(\boldsymbol{X}_t) - \boldsymbol{y})$。

选择 ϵ_t **和** α_t。

更新预条件子：

$$\boldsymbol{L}_t = \epsilon_t \boldsymbol{I}_{n_1} + \operatorname{diag}(\boldsymbol{G}_t \boldsymbol{G}_t^T).$$

$$\boldsymbol{R}_t = \epsilon_t \boldsymbol{I}_{n_2} + \operatorname{diag}(\boldsymbol{G}_t^T \boldsymbol{G}_t).$$

更新中间参数：

$$\boldsymbol{W}_t = \boldsymbol{X}_t - \alpha_t \widetilde{\mathcal{P}}_{\mathbb{T}_t}(\boldsymbol{L}_t^{-\frac{1}{4}} \boldsymbol{G}_t \boldsymbol{R}_t^{-\frac{1}{4}}).$$

更新 X：

$$\boldsymbol{X}_{t+1} = \mathcal{H}_r(\boldsymbol{W}_t).$$

满足停止条件时输出 $\boldsymbol{X}_{t+1}$。

2.2.3 PRGD中的计算。

与 RGD（2.4）类似，PRGD 的主要计算过程（如算法 1所述）涉及对以下内容的求值~投影算子 $\mathcal{P}_{\mathbb{T}_t}$ 和回缩算子 $\mathcal{H}_r$，以及对 $\boldsymbol{G}_t$ 的计算。通过利用~黎曼流形 $\mathcal{M}_r$的结构，可以高效地计算这两个算子 $\mathcal{P}_{\mathbb{T}_t}$ 和 $\mathcal{H}_r$。

- 投影的计算 $\mathcal{P}_{\mathbb{T}_t}$。回顾一下，$\widetilde{\mathcal{P}}_{\mathbb{T}_t}$ 表示关于 $\widetilde{\mathbb{R}^{n_1 \times n_2}}$的内积 $\langle \cdot, \cdot \rangle_{\mathcal{W}_t}$ 在切空间 $\mathbb{T}_t$上的投影。设 $\boldsymbol{X}_t = \boldsymbol{U}_t \boldsymbol{\Sigma}_t \boldsymbol{V}_t^T$为一个紧奇异值分解，其中$\boldsymbol{U}_t = [\boldsymbol{u}_t^1, \boldsymbol{u}_t^2, \ldots, \boldsymbol{u}_t^r] \in \mathbb{R}^{n_1 \times r}$且 $\boldsymbol{V}_t = [\boldsymbol{v}_t^1, \boldsymbol{v}_t^2, \ldots, \boldsymbol{v}_t^r] \in \mathbb{R}^{n_2 \times r}$。我们定义加权内积

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle_{\boldsymbol{L}_t^{\frac{1}{4}}} = \langle \boldsymbol{L}_t^{\frac{1}{4}} \boldsymbol{x}, \boldsymbol{y} \rangle \quad \text{in} \quad \mathbb{R}^{n_1} \quad \text{and} \quad \langle \boldsymbol{x}, \boldsymbol{y} \rangle_{\boldsymbol{R}_t^{\frac{1}{4}}} = \langle \boldsymbol{R}_t^{\frac{1}{4}} \boldsymbol{x}, \boldsymbol{y} \rangle \quad \text{in} \quad \mathbb{R}^{n_2}, \tag{2.11}$$

并在这些内积下对 $\{\boldsymbol{u}_t^i\}_{i=1}^r$ 和 $\{\boldsymbol{v}_t^i\}_{i=1}^r$ 进行正交化，以获得标准正交基

$$\widetilde{\boldsymbol{U}}_t = \boldsymbol{U}_t \big(\boldsymbol{U}_t^T \boldsymbol{L}_t^{\frac{1}{4}} \boldsymbol{U}_t\big)^{-\frac{1}{2}} := \big[\tilde{\boldsymbol{u}}_t^1, \tilde{\boldsymbol{u}}_t^2, \ldots, \tilde{\boldsymbol{u}}_t^r\big] \in \mathbb{R}^{n_1 \times r},$$

$$\widetilde{\boldsymbol{V}}_t = \boldsymbol{V}_t \big(\boldsymbol{V}_t^T \boldsymbol{R}_t^{\frac{1}{4}} \boldsymbol{V}_t\big)^{-\frac{1}{2}} := \big[\tilde{\boldsymbol{v}}_t^1, \tilde{\boldsymbol{v}}_t^2, \ldots, \tilde{\boldsymbol{v}}_t^r\big] \in \mathbb{R}^{n_2 \times r}.$$

~ ~我们进一步将 $\boldsymbol{U}_t$和 $\boldsymbol{V}_t$分别扩展为（$\mathbb{R}^{n_1}$, $\langle \cdot, \cdot \rangle_{\boldsymbol{L}_t^{\frac{1}{4}}}$）和（$\mathbb{R}^{n_2}$, $\langle \cdot, \cdot \rangle_{\boldsymbol{R}_t^{\frac{1}{4}}}$）的完整标准正交基。然后，关于 $\langle \cdot, \cdot \rangle_{\mathcal{W}_t}$的 $\mathbb{T}_t$的一个标准正交基可表示为 $\big\{\tilde{\boldsymbol{u}}_t^i (\tilde{\boldsymbol{v}}_t^j)^T\big\}_{\min\{i,j\} \le r}$。利用~这个标准正交基，我们可以直接计算投影 $\mathcal{P}_{\mathbb{T}_t}$ 到 $\mathbb{T}_t$ 上。具体而言，对于任意

$\boldsymbol{Z} \in \mathbb{R}^{n_1 \times n_2}$，我们有

$$\begin{aligned}\widetilde{\mathcal{P}}_{\mathbb{T}_t}\boldsymbol{Z} &= \sum_{(i,j):\min\{i,j\}\le r} \langle \boldsymbol{Z}, \tilde{\boldsymbol{u}}_t^i(\tilde{\boldsymbol{v}}_t^j)^T\rangle_{\mathcal{W}_t} \cdot \tilde{\boldsymbol{u}}_t^i(\tilde{\boldsymbol{v}}_t^j)^T = \sum_{(i,j):\min\{i,j\}\le r} \langle \boldsymbol{Z}, \boldsymbol{L}_t^{\frac{1}{4}}\tilde{\boldsymbol{u}}_t^i(\tilde{\boldsymbol{v}}_t^j)^T\boldsymbol{R}_t^{\frac{1}{4}}\rangle \cdot \tilde{\boldsymbol{u}}_t^i(\tilde{\boldsymbol{v}}_t^j)^T \\ &= \sum_{(i,j):\min\{i,j\}\le r} (\tilde{\boldsymbol{u}}_t^i)^T\boldsymbol{L}_t^{\frac{1}{4}}\boldsymbol{Z}\boldsymbol{R}_t^{\frac{1}{4}}\tilde{\boldsymbol{v}}_t^j \cdot \tilde{\boldsymbol{u}}_t^i(\tilde{\boldsymbol{v}}_t^j)^T \\ &= \widetilde{\boldsymbol{U}}_t\widetilde{\boldsymbol{U}}_t^T\boldsymbol{L}_t^{\frac{1}{4}}\boldsymbol{Z} + \boldsymbol{Z}\boldsymbol{R}_t^{\frac{1}{4}}\widetilde{\boldsymbol{V}}_t\widetilde{\boldsymbol{V}}_t^T - \widetilde{\boldsymbol{U}}_t\widetilde{\boldsymbol{U}}_t\boldsymbol{L}_t^{\frac{1}{4}}\boldsymbol{Z}\boldsymbol{R}_t^{\frac{1}{4}}\widetilde{\boldsymbol{V}}_t\widetilde{\boldsymbol{V}}_t^T.\end{aligned} \tag{2.12}$$

- 截断奇异值分解的计算 $\mathcal{H}_r$。在算法 1中，很明显 $\boldsymbol{W}_t \in \mathbb{T}_t$。利用切空间 $\mathbb{T}_t$ 在（2.5）中的形式，我们可以看到 $\boldsymbol{W}_t$ 的秩至多为 $2r$。这使我们能够高效地计算 $\mathcal{H}_r(\boldsymbol{W}_t)$ 中的算法 1 ，而无需进行大规模奇异值分解。为此，我们可以直接计算 $\boldsymbol{W}_t$ 如下：

$$\begin{aligned}\boldsymbol{W}_t &= \boldsymbol{X}_t + \widetilde{\mathcal{P}}_{\mathbb{T}_t}(\boldsymbol{Z}_t) \\ &= \boldsymbol{U}_t\boldsymbol{\Sigma}_t\boldsymbol{V}_t^T + \widetilde{\boldsymbol{U}}_t\widetilde{\boldsymbol{U}}_t^T\boldsymbol{L}_t^{\frac{1}{4}}\boldsymbol{Z}_t + \boldsymbol{Z}_t\boldsymbol{R}_t^{\frac{1}{4}}\widetilde{\boldsymbol{V}}_t\widetilde{\boldsymbol{V}}_t^T - \widetilde{\boldsymbol{U}}_t\widetilde{\boldsymbol{U}}_t^T\boldsymbol{L}_t^{\frac{1}{4}}\boldsymbol{Z}_t\boldsymbol{R}_t^{\frac{1}{4}}\widetilde{\boldsymbol{V}}_t\widetilde{\boldsymbol{V}}_t^T \\ &= \boldsymbol{U}_t\boldsymbol{\Sigma}_t\boldsymbol{V}_t^T + \boldsymbol{U}_t\big(\underbrace{\boldsymbol{U}_t^T\boldsymbol{L}_t^{\frac{1}{4}}\boldsymbol{U}_t}_{:=\boldsymbol{M}_1}\big)^{-1}\boldsymbol{U}_t^T\boldsymbol{L}_t^{\frac{1}{4}}\boldsymbol{Z}_t + \boldsymbol{Z}_t\boldsymbol{R}_t^{\frac{1}{4}}\boldsymbol{V}_t\big(\underbrace{\boldsymbol{V}_t^T\boldsymbol{R}_t^{\frac{1}{4}}\boldsymbol{V}}_{:=\boldsymbol{M}_2}\big)^{-1}\boldsymbol{V}_t^T \\ &\quad - \boldsymbol{U}_t\big(\boldsymbol{U}_t^T\boldsymbol{L}_t^{\frac{1}{4}}\boldsymbol{U}_t\big)^{-1}\boldsymbol{U}_t^T\boldsymbol{L}_t^{\frac{1}{4}}\boldsymbol{Z}_t\boldsymbol{R}_t^{\frac{1}{4}}\boldsymbol{V}_t\big(\boldsymbol{V}_t^T\boldsymbol{R}_t^{\frac{1}{4}}\boldsymbol{V}_t\big)^{-1}\boldsymbol{V}_t^T \\ &= \boldsymbol{U}_t\big(\underbrace{\boldsymbol{\Sigma}_t + \boldsymbol{M}_1^{-1}\boldsymbol{U}_t^T\boldsymbol{L}_t^{\frac{1}{4}}\boldsymbol{Z}_t\boldsymbol{V}_t + \big(\boldsymbol{U}_t^T - \boldsymbol{M}_1^{-1}\boldsymbol{U}_t^T\boldsymbol{L}_t^{\frac{1}{4}}\big)\boldsymbol{Z}_t\boldsymbol{R}_t^{\frac{1}{4}}\boldsymbol{V}_t\boldsymbol{M}_2^{-1}}_{:=\boldsymbol{K}_0}\big)\boldsymbol{V}_t^T \\ &\quad + \boldsymbol{U}_t\underbrace{\boldsymbol{M}_1^{-1}\boldsymbol{U}_t^T\boldsymbol{L}_t^{\frac{1}{4}}\boldsymbol{Z}_t\left(\boldsymbol{I} - \boldsymbol{V}_t\boldsymbol{V}_t^T\right)}_{:=\boldsymbol{Y}_1^T} + \underbrace{\left(\boldsymbol{I} - \boldsymbol{U}_t\boldsymbol{U}_t^T\right)\boldsymbol{Z}_t\boldsymbol{R}_t^{\frac{1}{4}}\boldsymbol{V}_t\boldsymbol{M}_2^{-1}}_{:=\boldsymbol{Y}_2}\boldsymbol{V}_t^T\end{aligned}$$

然后，我们可以对矩阵 $\boldsymbol{Y}_1$ 和 $\boldsymbol{Y}_2$ 进行 QR分解，分别得到 $\boldsymbol{Y}_1 = \boldsymbol{Q}_1\boldsymbol{K}_1$ 和 $\boldsymbol{Y}_2 = \boldsymbol{Q}_2\boldsymbol{K}_2$。显然，$\boldsymbol{U}_t^T\boldsymbol{Q}_2 = \boldsymbol{0}$ 和 $\boldsymbol{V}_t^T\boldsymbol{Q}_1 = \boldsymbol{0}$。利用所得的因子分解，我们可以将 $\boldsymbol{W}_t$ 重写如下：

$$\boldsymbol{W}_t = \boldsymbol{U}_t\boldsymbol{K}_0\boldsymbol{V}_t^T + \boldsymbol{U}_t\boldsymbol{K}_1^T\boldsymbol{Q}_1^T + \boldsymbol{Q}_2\boldsymbol{K}_2\boldsymbol{V}_t^T = \begin{bmatrix}\boldsymbol{U}_t & \boldsymbol{Q}_2\end{bmatrix}\underbrace{\begin{bmatrix}\boldsymbol{K}_0 & \boldsymbol{K}_1^T \\ \boldsymbol{K}_2 & \boldsymbol{0}\end{bmatrix}}_{\boldsymbol{M}_t}\begin{bmatrix}\boldsymbol{V}_t^T \\ \boldsymbol{Q}_1^T\end{bmatrix},$$

其中 $\boldsymbol{M}_t$ 是 $2r \times 2r$ 矩阵。由于 $\begin{bmatrix}\boldsymbol{U}_t & \boldsymbol{Q}_2\end{bmatrix}$ 和 $\begin{bmatrix}\boldsymbol{V}_t & \boldsymbol{Q}_1\end{bmatrix}$ 都是正交矩阵，因此可以通过对 $\boldsymbol{M}_t$ 进行奇异值分解来得到 $\boldsymbol{W}_t$ 的奇异值分解，而该分解可使用 $O(r^3)$ 次浮点运算次数计算得到。因此，无需进行任何大规模奇异值分解，即可高效地完成截断奇异值分解 $\mathcal{H}_r(\boldsymbol{W}_t)$ 。

与采用典范度量的 RGD（2.4）的计算相比，所提出的 PRGD 需要额外计算矩阵 $\boldsymbol{L}_t$、$\boldsymbol{R}_t$、$\boldsymbol{K}_1^{-1}$和 $\boldsymbol{K}_2^{-1}$，这些矩阵要么是对角矩阵，要么尺寸较小。因此，PRGD 仅带来极少的额外计算。正如我们稍后将展示的，与 RGD 相比，PRGD 能显著减少所需的迭代次数。总之，PRGD 是一种效率远高于 RGD 的算法。

3 恢复保证

在本节中，我们为 PRGD 算法（即算法 1）提供恢复保证，并假设感知算子 $\mathcal{A}$ 满足受限等距性质（RIP） [6]。分析过程涉及将 [36] 中针对典范度量下 RGD 的证明进行修改，使其适用于 PRGD 中使用的非典范度量。当 RIP

<escape>不满足时，仍可结合本文中的技术与相应证明，建立 PRGD 的恢复保证理论 [26, 35]，即采用典范度量的 RGD 的相关证明。</escape>

低秩矩阵恢复中的 RIP 首次在 [6] 中提出。它可以追溯到 [14], ，其中建立了压缩感知中的 RIP。对于低秩矩阵恢复，RIP 定义如下。

定义 3.1（受限等距性质（RIP））。线性算子 $\mathcal{A}$： $\mathbb{R}^{n_1\times n_2}\to\mathbb{R}^m$ 满足 s 阶受限等距性质，当且仅当存在常数 $\delta_s\in$ （0，1）使得

$$(1-\delta_s)\|\boldsymbol{Z}\|_F^2\le\|\mathcal{A}\boldsymbol{Z}\|_2^2\le(1+\delta_s)\|\boldsymbol{Z}\|_F^2,\qquad\forall\ \boldsymbol{Z}\ :\ \operatorname{rank}(\boldsymbol{Z})\le s. \tag{3.1}$$

当感知矩阵 $\boldsymbol{A}_i$， $1\le i\le m$在（1.2）中包含服从均值为 0、方差为 $\frac{1}{m}$ 的次高斯分布的独立同分布随机元素时，只要满足$m\ge C\left(n_1+n_2\right)s\log\left(n_1n_2\right)$，感知算子 $\mathcal{A}$ 就以高概率满足 RIP，其中 C 取决于 δ_s。RIP 已被广泛用于分析低秩矩阵恢复的凸和非凸方法。在 RIP 的假设下，核范数最小化可以精确恢复底层低秩矩阵 [6]。在我们之前的工作 [36], 中，我们证明了，当 $\mathcal{A}$ 满足 RIP 时，RGD（2.4）从 $\boldsymbol{X}$ 出发，经由一步 IHT 初始化后即可0精确恢复。还有许多其他结果表明，RIP 可以通过奇异值投影、因子梯度下降、交替最小化、商流形上的高斯-牛顿方法以及全局景观分析等方法实现精确的低秩矩阵恢复。

借助 RIP，我们现在在以下定理中给出主要结果。

定理 3.2（PRGD的恢复保证）。假设 $\mathcal{A}$ 满足阶数为 $3r$ 、常数为δ_{3r}的 *RIP*。设 $\boldsymbol{X}\in\mathbb{R}^{n_1\times n_2}$ 的秩为 r，条件数为 κ。令输入数据 $\boldsymbol{y}=\mathcal{A}\boldsymbol{X}$ 和初始猜测$\boldsymbol{X}_0=\mathcal{H}_r\left(\mathcal{A}^*\left(\boldsymbol{y}\right)\right)$ 取自算法*1*，并选择 α_t， $\epsilon_t>0$ 为根据 $\boldsymbol{G}_t$ 对所有 $t\in\mathbb{N}$计算得到的一些正参数。于是，若

$$\delta_{3r}\le\frac{1}{50+100\sqrt{r}\kappa},$$

序列 $\{\boldsymbol{X}_t\}_{t\in\mathbb{N}}$ 由算法生成*1* 满足

$$\|\boldsymbol{X}_{t+1}-\boldsymbol{X}\|_F\le\mu\|\boldsymbol{X}_t-\boldsymbol{X}\|_F,$$

其中， $\mu\le0.995$ 是一个正数常量。

定理 3.2 表明，RIP 常数 $\delta_{3r}\lesssim\frac{1}{\sqrt{r}\kappa}$ 是 PRGD 精确恢复的充分条件。当感知矩阵的元素服从均值为 0、方差为 $\frac{1}{m}$ 的独立同分布次高斯分布时，只要满足 $m\gtrsim\kappa^2r^2\max\{n_1,n_2\}$，即可满足对 RIP 常数的这一要求。因此，PRGD 精确恢复所需的样本数为 $O(\kappa^2r^2\max\{n_1,n_2\})$，这在矩阵维数方面是最优的，且与 RGD 和迭代硬阈值等许多非凸算法所需的样本数同阶。

本节其余部分给出定理 3.2 的证明。首先，我们在第 3.1 节中介绍一些辅助引理，随后在第 3.2 节给出主要证明。

3.1 辅助引理

我们的分析基于加权内积 $\langle\cdot,\cdot\rangle_{\mathcal{W}_t}$ （2.10）中定义的加权内积及其诱导范数 $\|\cdot\|_{\mathcal{W}_t}$。在分析中，我们需要一些与该加权内积和范数相关的基本不等式，它们在我们的分析中起着关键作用。

我们首先在以下引理中建立加权范数 $\|\cdot\|_{\mathcal{W}_t}$与标准范数 $\|\cdot\|_F$之间的等价关系。

引理 3.3。对于任意 $t\in\mathbb{N}$ 和 $\boldsymbol{Z}\in\mathbb{R}^{n_1\times n_2}$，我们有

$$\nu_t\|\boldsymbol{Z}\|_F^2\le\|\boldsymbol{Z}\|_{\mathcal{W}_t}^2\le\mu_t\|\boldsymbol{Z}\|_F^2, \tag{3.2}$$

其中 $\nu_t=\epsilon_t^{\frac{1}{2}}$且 $\mu_t=\left(\epsilon_t+\|\boldsymbol{G}_t\|_\vee^2\right)^{\frac{1}{2}}$满足 $\|\cdot\|_\vee$定义见 (1.7)。

证明。设 $l_{t,\,i}$ 和 $r_{t,\,i}$ 分别为 $\boldsymbol{L}_t$ 和 $\boldsymbol{R}_t$的第 i 个对角元素。根据 $\boldsymbol{L}_t = \epsilon_t \boldsymbol{I} + \mathrm{diag}(\boldsymbol{G}_t \boldsymbol{G}_t^T)$ 和 $\boldsymbol{R}_t = \epsilon_t \boldsymbol{I} + \mathrm{diag}(\boldsymbol{G}_t^T \boldsymbol{G}_t)$ 的定义，有

$$l_{t,i} = \epsilon_t + \|\boldsymbol{G}_t(i,:)\|_2^2, \quad r_{t,i} = \epsilon_t + \|\boldsymbol{G}_t(:,i)\|_2^2, \qquad \forall\, i,$$

其中，$\boldsymbol{G}_t(i,:)$ 和 $\boldsymbol{G}_t(:,i)$ 分别是 $\boldsymbol{G}_t$ 的第 i 行和第 i 列。此外，根据 $\|\cdot\|_{\mathcal{W}_t}$ 的定义，我们知道

$$\|\boldsymbol{Z}\|_{\mathcal{W}_t}^2 = \left\langle \boldsymbol{L}_t^{\frac{1}{4}} \boldsymbol{Z} \boldsymbol{R}_t^{\frac{1}{4}}, \boldsymbol{Z} \right\rangle = \sum_{i,j} l_{t,i}^{\frac{1}{4}} r_{t,j}^{\frac{1}{4}} Z_{ij}^2 \quad \text{for any} \quad \boldsymbol{Z} \in \mathbb{R}^{n_1 \times n_2}. \tag{3.3}$$

为了得到 ν_t 和 μ_t 在（3.2）中，我们只需找到所有 i 对应的 $l_{t,i}$ 和 $r_{t,i}$ 的下界和上界。

- 对于下界 ν_t，我们使用平凡下界 $l_{t,i} \geq \epsilon_t$ 和 $r_{t,i} \geq \epsilon_t$，对所有 i 均成立。因此，由（3.3），可得

$$\|\boldsymbol{Z}\|_{\mathcal{W}_t}^2 \geq \epsilon_t^{\frac{1}{2}} \|\boldsymbol{Z}\|_F^2.$$

- 对于上界 μ_t，我们使用

$$\max\{l_{t,i}, r_{t,i}\} \leq \epsilon_t + \|\boldsymbol{G}_t\|_\vee^2, \quad \forall\, i.$$

  将其代入（3.3），可得

$$\|\boldsymbol{Z}\|_{\mathcal{W}_t}^2 \leq \left(\epsilon_t + \|\boldsymbol{G}_t\|_\vee^2\right)^{\frac{1}{2}} \|\boldsymbol{Z}\|_F^2.$$

□

为了完成收敛性证明，我们需要估计加权内积 $\langle\cdot,\cdot\rangle_{\boldsymbol{L}_t^{\frac{1}{4}}}$和范数 $\|\cdot\|_{\boldsymbol{L}_t^{\frac{1}{4}}}$(以及类似地，$\langle\cdot,\cdot\rangle_{\boldsymbol{R}_t^{\frac{1}{4}}}$和 $\|\cdot\|_{\boldsymbol{R}_t^{\frac{1}{4}}}$)在 $\mathbb{R}^{n_1}$中的值，定义见（2.11）。我们引入矩阵算子范数 $\|\boldsymbol{A}\|_{\boldsymbol{L}_t^{\frac{1}{4}}} = \sup_{\|\boldsymbol{x}\|\neq 0} \frac{\|\boldsymbol{A}\boldsymbol{x}\|_{\boldsymbol{L}_t^{\frac{1}{4}}}}{\|\boldsymbol{x}\|_{\boldsymbol{L}_t^{\frac{1}{4}}}}$，适用于 $\mathbb{R}^{n_1 \times n_1}$(以及类似地，$\|\cdot\|_{\boldsymbol{R}_t^{\frac{1}{4}}}$)。下面给出关于这两种范数的两个引理。

引理 3.4。设 $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{n_1 \times n_1}$。则有

$$\|\boldsymbol{A}\boldsymbol{B}\|_{\boldsymbol{L}_t^{\frac{1}{4}}} \leq \mathrm{cond}(\boldsymbol{L}_t^{\frac{1}{8}}) \cdot \|\boldsymbol{A}\|_{\boldsymbol{L}_t^{\frac{1}{4}}} \|\boldsymbol{B}\|_2.$$

证明。根据 $\|\cdot\|_{\boldsymbol{L}_t^{\frac{1}{4}}}$ 的定义，可得

$$\|\boldsymbol{A}\|_{\boldsymbol{L}_t^{\frac{1}{4}}} = \sup_{\|\boldsymbol{x}\|\neq 0} \frac{\|\boldsymbol{A}\boldsymbol{x}\|_{\boldsymbol{L}_t^{\frac{1}{4}}}}{\|\boldsymbol{x}\|_{\boldsymbol{L}_t^{\frac{1}{4}}}} = \sup_{\|\boldsymbol{x}\|\neq 0} \frac{\|\boldsymbol{L}_t^{\frac{1}{8}} \boldsymbol{A}\boldsymbol{x}\|_2}{\|\boldsymbol{L}_t^{\frac{1}{8}} \boldsymbol{x}\|_2} = \sup_{\|\boldsymbol{y}\|\neq 0} \frac{\|\boldsymbol{L}_t^{\frac{1}{8}} \boldsymbol{A} \boldsymbol{L}_t^{-\frac{1}{8}} \boldsymbol{y}\|_2}{\|\boldsymbol{y}\|_2} = \|\boldsymbol{L}_t^{\frac{1}{8}} \boldsymbol{A} \boldsymbol{L}_t^{-\frac{1}{8}}\|_2.$$

因此，直接计算可得

$$\|\boldsymbol{A}\boldsymbol{B}\|_{\boldsymbol{L}_t^{\frac{1}{4}}} = \|\boldsymbol{L}_t^{\frac{1}{8}} \boldsymbol{A}\boldsymbol{B}\boldsymbol{L}_t^{-\frac{1}{8}}\|_2 \leq \|\boldsymbol{L}_t^{\frac{1}{8}} \boldsymbol{A} \boldsymbol{L}_t^{-\frac{1}{8}}\|_2 \|\boldsymbol{L}_t^{\frac{1}{8}}\|_2 \|\boldsymbol{B}\|_2 \|\boldsymbol{L}_t^{-\frac{1}{8}}\|_2 = \mathrm{cond}(\boldsymbol{L}_t^{\frac{1}{8}}) \cdot \|\boldsymbol{A}\|_{\boldsymbol{L}_t^{\frac{1}{4}}} \|\boldsymbol{B}\|_2.$$

证明完毕。 □

引理 3.5。设 $\boldsymbol{A} \in \mathbb{R}^{n_1 \times n_1}$、$\boldsymbol{B} \in \mathbb{R}^{n_1 \times n_2}$ 和 $\boldsymbol{C} \in \mathbb{R}^{n_2 \times n_2}$。那么，我们有

$$\|\boldsymbol{A}\boldsymbol{B}\|_{\mathcal{W}_t} \leq \|\boldsymbol{A}\|_{\boldsymbol{L}_t^{\frac{1}{4}}} \|\boldsymbol{B}\|_{\mathcal{W}_t} \quad \textit{and} \quad \|\boldsymbol{B}\boldsymbol{C}\|_{\mathcal{W}_t} \leq \mathrm{cond}(\boldsymbol{R}_t^{-\frac{1}{4}}) \cdot \|\boldsymbol{B}\|_{\mathcal{W}_t} \|\boldsymbol{C}\|_{\boldsymbol{R}_t^{\frac{1}{4}}}. \tag{3.4}$$

证明。 利用 $\|\cdot\|_{\mathcal{W}_t}$ 的定义和弗罗贝尼乌斯范数的一个性质，我们得到

$$\|\boldsymbol{AB}\|_{\mathcal{W}_t} = \|\boldsymbol{L}_t^{\frac{1}{8}}\boldsymbol{AB}\boldsymbol{R}_t^{\frac{1}{8}}\|_F \le \|\boldsymbol{L}_t^{\frac{1}{8}}\boldsymbol{A}\boldsymbol{L}_t^{-\frac{1}{8}}\|_2\|\boldsymbol{L}_t^{\frac{1}{8}}\boldsymbol{B}\boldsymbol{R}_t^{\frac{1}{8}}\|_F = \|\boldsymbol{L}_t^{\frac{1}{8}}\boldsymbol{A}\boldsymbol{L}_t^{-\frac{1}{8}}\|_2\|\boldsymbol{B}\|_{\mathcal{W}_t} = \|\boldsymbol{A}\|_{\boldsymbol{L}_t^{\frac{1}{4}}}\|\boldsymbol{B}\|_{\mathcal{W}_t},$$

这是（3.4）中的第一个不等式。对于第二个不等式，我们注意到

$$\|\boldsymbol{BC}\|_{\mathcal{W}_t} = \|\boldsymbol{L}_t^{\frac{1}{8}}\boldsymbol{BC}\boldsymbol{R}_t^{\frac{1}{8}}\|_F \le \|\boldsymbol{B}\|_{\mathcal{W}_t}\|\boldsymbol{R}_t^{-\frac{1}{8}}\boldsymbol{C}\boldsymbol{R}_t^{\frac{1}{8}}\|_2.$$

1 1记 $\boldsymbol{C}_0 = \boldsymbol{R}_t^{\frac{1}{8}}\boldsymbol{C}\boldsymbol{R}_t^{-\frac{1}{8}}$，我们有 $\|\boldsymbol{C}_0\|_2 = \|\boldsymbol{C}\|_{\boldsymbol{R}_t^{\frac{1}{4}}}$且

$$\|\boldsymbol{R}_t^{-\frac{1}{8}}\boldsymbol{C}\boldsymbol{R}_t^{\frac{1}{8}}\|_2 = \|\boldsymbol{R}_t^{-\frac{1}{4}}\boldsymbol{C}_0\boldsymbol{R}_t^{\frac{1}{4}}\|_2 \le \|\boldsymbol{R}_t^{-\frac{1}{4}}\|_2\|\boldsymbol{C}_0\|_2\|\boldsymbol{R}_t^{\frac{1}{4}}\|_2 = \mathrm{cond}(\boldsymbol{R}_t^{-\frac{1}{4}})\cdot\|\boldsymbol{C}_0\|_2.$$

因此，我们得到

$$\|\boldsymbol{BC}\|_{\mathcal{W}_t} \le \mathrm{cond}(\boldsymbol{R}_t^{-\frac{1}{4}})\cdot\|\boldsymbol{B}\|_{\mathcal{W}_t}\|\boldsymbol{R}_t^{\frac{1}{8}}\boldsymbol{C}\boldsymbol{R}_t^{-\frac{1}{8}}\|_2 = \mathrm{cond}(\boldsymbol{R}_t^{-\frac{1}{4}})\cdot\|\boldsymbol{B}\|_{\mathcal{W}_t}\|\boldsymbol{C}\|_{\boldsymbol{R}_t^{\frac{1}{4}}}.$$

□

基于上述不等式，我们可以对两个矩阵的列子空间给出如下估计：

引理 3.6。 设 $\boldsymbol{X}_t = \boldsymbol{U}_t\boldsymbol{\Sigma}_t\boldsymbol{V}_t^T$ 和 $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ 分别是 $\boldsymbol{X}_t$ 和 $\boldsymbol{X}$ 的紧奇异值分解。则有

$$\|\widetilde{\boldsymbol{U}}_t\widetilde{\boldsymbol{U}}_t^T\boldsymbol{L}_t^{\frac{1}{4}} - \widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{U}}^T\boldsymbol{L}_t^{\frac{1}{4}}\|_{\boldsymbol{L}_t^{\frac{1}{4}}} \le \frac{\mathrm{cond}(\boldsymbol{L}_t^{\frac{3}{8}})}{\sigma_{\min}(\boldsymbol{X})}\|\boldsymbol{X}_t - \boldsymbol{X}\|_2, \tag{3.5}$$

其中 $\widetilde{\boldsymbol{U}}_t = \boldsymbol{U}_t\big(\boldsymbol{U}_t^T\boldsymbol{L}_t^{\frac{1}{4}}\boldsymbol{U}_t\big)^{-\frac{1}{2}}$ 和 $\widetilde{\boldsymbol{U}} = \boldsymbol{U}\big(\boldsymbol{U}^T\boldsymbol{L}_t^{\frac{1}{4}}\boldsymbol{U}\big)^{-\frac{1}{2}}$ 分别是在内积 $\langle\cdot,\cdot\rangle_{\boldsymbol{L}_t}$ 下对 $\boldsymbol{U}_t$ 和 $\boldsymbol{U}$ 正交归一化得到的。

证明。 我们沿用 [36,引理 4。2]中的相同论证。由于 $\widetilde{\boldsymbol{U}}_t\widetilde{\boldsymbol{U}}_t^T\boldsymbol{L}_t^{\frac{1}{4}}$和 $\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{U}}^T\boldsymbol{L}_t^{\frac{1}{4}}$是关于内积 $\langle\cdot,\cdot\rangle_{\boldsymbol{L}_t^{\frac{1}{4}}}$分别投影到 $\boldsymbol{X}_t$和 $\boldsymbol{X}$的列子空间上的正交投影， [21,定理 2.6。1] 可得

$$\big\|\widetilde{\boldsymbol{U}}_t\widetilde{\boldsymbol{U}}_t^T\boldsymbol{L}_t^{\frac{1}{4}} - \widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{U}}^T\boldsymbol{L}_t^{\frac{1}{4}}\big\|_{\boldsymbol{L}_t^{\frac{1}{4}}} = \big\|\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{U}}^T\boldsymbol{L}_t^{\frac{1}{4}}\big(\boldsymbol{I} - \widetilde{\boldsymbol{U}}_t\widetilde{\boldsymbol{U}}_t^T\boldsymbol{L}_t^{\frac{1}{4}}\big)\big\|_{\boldsymbol{L}_t^{\frac{1}{4}}} = \big\|\big(\boldsymbol{I} - \widetilde{\boldsymbol{U}}_t\widetilde{\boldsymbol{U}}_t^T\boldsymbol{L}_t^{\frac{1}{4}}\big)\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{U}}^T\boldsymbol{L}_t^{\frac{1}{4}}\big\|_{\boldsymbol{L}_t^{\frac{1}{4}}}.$$

~ ~利用 $\boldsymbol{U}$ 和 $\boldsymbol{U}_t$ 的定义，我们有

$$\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{U}}^T = \boldsymbol{U}\big(\boldsymbol{U}^T\boldsymbol{L}_t^{\frac{1}{4}}\boldsymbol{U}\big)^{-1}\boldsymbol{U}^T = \boldsymbol{X}\boldsymbol{V}\boldsymbol{\Sigma}^{-1}\big(\boldsymbol{U}^T\boldsymbol{L}_t^{\frac{1}{4}}\boldsymbol{U}\big)^{-1}\boldsymbol{U}^T$$

and

$$\big(\boldsymbol{I} - \widetilde{\boldsymbol{U}}_t\widetilde{\boldsymbol{U}}_t^T\boldsymbol{L}_t^{\frac{1}{4}}\big)\boldsymbol{X}_t = \boldsymbol{0}.$$

综上，我们得到

$$\begin{aligned}
\big\|\widetilde{\boldsymbol{U}}_t\widetilde{\boldsymbol{U}}_t^T\boldsymbol{L}_t^{\frac{1}{4}} - \widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{U}}^T\boldsymbol{L}_t^{\frac{1}{4}}\big\|_{\boldsymbol{L}_t^{\frac{1}{4}}} &= \big\|\big(\boldsymbol{I} - \widetilde{\boldsymbol{U}}_t\widetilde{\boldsymbol{U}}_t^T\boldsymbol{L}_t^{\frac{1}{4}}\big)\boldsymbol{X}\boldsymbol{V}\boldsymbol{\Sigma}^{-1}\big(\boldsymbol{U}^T\boldsymbol{L}_t^{\frac{1}{4}}\boldsymbol{U}\big)^{-1}\boldsymbol{U}^T\boldsymbol{L}_t^{\frac{1}{4}}\big\|_{\boldsymbol{L}_t^{\frac{1}{4}}} \\
&= \big\|\big(\boldsymbol{I} - \widetilde{\boldsymbol{U}}_t\widetilde{\boldsymbol{U}}_t^T\boldsymbol{L}_t^{\frac{1}{4}}\big)(\boldsymbol{X} - \boldsymbol{X}_t)\boldsymbol{V}\boldsymbol{\Sigma}^{-1}\big(\boldsymbol{U}^T\boldsymbol{L}_t^{\frac{1}{4}}\boldsymbol{U}\big)^{-1}\boldsymbol{U}^T\boldsymbol{L}_t^{\frac{1}{4}}\big\|_{\boldsymbol{L}_t^{\frac{1}{4}}} \\
&\le \mathrm{cond}(\boldsymbol{L}_t^{\frac{1}{8}})\cdot\|\boldsymbol{I} - \widetilde{\boldsymbol{U}}_t\widetilde{\boldsymbol{U}}_t^T\boldsymbol{L}_t^{\frac{1}{4}}\|_{\boldsymbol{L}_t^{\frac{1}{4}}}\|(\boldsymbol{X} - \boldsymbol{X}_t)\boldsymbol{V}\boldsymbol{\Sigma}^{-1}(\boldsymbol{U}^T\boldsymbol{L}_t^{\frac{1}{4}}\boldsymbol{U})^{-1}\boldsymbol{U}^T\boldsymbol{L}_t^{\frac{1}{4}}\|_2 \\
&\le \mathrm{cond}(\boldsymbol{L}_t^{\frac{1}{8}})\cdot\|\boldsymbol{X} - \boldsymbol{X}_t\|_2\|\boldsymbol{V}\|_2\|\boldsymbol{\Sigma}^{-1}\|_2\big\|\big(\boldsymbol{U}^T\boldsymbol{L}_t^{\frac{1}{4}}\boldsymbol{U}\big)^{-1}\big\|_2\|\boldsymbol{U}^T\|_2\|\boldsymbol{L}_t^{\frac{1}{4}}\|_2 \\
&\le \mathrm{cond}(\boldsymbol{L}_t^{\frac{1}{8}})\cdot\frac{\|\boldsymbol{X} - \boldsymbol{X}_t\|_2}{\sigma_{\min}(\boldsymbol{X})}\frac{\lambda_{\max}(\boldsymbol{L}_t^{\frac{1}{4}})}{\lambda_{\min}(\boldsymbol{L}_t^{\frac{1}{4}})} = \frac{\mathrm{cond}(\boldsymbol{L}_t^{\frac{3}{8}})}{\sigma_{\min}(\boldsymbol{X})}\|\boldsymbol{X} - \boldsymbol{X}_t\|_2,
\end{aligned}$$

其中，在这三个不等式中，我们分别使用了引理 3.4、$\boldsymbol{I}-\widetilde{\boldsymbol{U}}_t\widetilde{\boldsymbol{U}}_t^T\boldsymbol{L}_t^{\frac{1}{4}}$ 是加权内积下的正交投影算子这一事实，以及由此得到的不等式 $\|(\boldsymbol{U}^T\boldsymbol{L}_t^{\frac{1}{4}}\boldsymbol{U})^{-1}\|_2 \le \frac{1}{\lambda_{\min}(\boldsymbol{L}_t^{\frac{1}{4}})}$。

$$\|(\boldsymbol{U}^T\boldsymbol{L}_t^{\frac{1}{4}}\boldsymbol{U})^{-1}\|_2=\frac{1}{\min_{\|\boldsymbol{x}\|_2=1}\langle\boldsymbol{U}^T\boldsymbol{L}_t^{\frac{1}{4}}\boldsymbol{U}\boldsymbol{x},\boldsymbol{x}\rangle}\le\frac{1}{\lambda_{\min}(\boldsymbol{L}_t^{\frac{1}{4}})\cdot\min_{\|\boldsymbol{x}\|=1}\langle\boldsymbol{U}\boldsymbol{x},\boldsymbol{U}\boldsymbol{x}\rangle}=\frac{1}{\lambda_{\min}(\boldsymbol{L}_t^{\frac{1}{4}})}.$$

□

下面的引理给出了投影的重要性质 $\mathcal{P}_{\mathbb{T}_t}$，该性质包含了光滑低秩矩阵流形的二阶信息，如 [36]所示。此引理对于证明我们的主要结果至关重要。

引理 3.7。我们有

$$\|(\mathcal{I}-\widetilde{\mathcal{P}}_{\mathbb{T}_t})\boldsymbol{X}\|_{\mathcal{W}_t}\le\frac{\mu_t^{\frac{5}{4}}}{\nu_t^{\frac{7}{4}}\cdot\sigma_{\min}(\boldsymbol{X})}\|\boldsymbol{X}-\boldsymbol{X}_t\|_{\mathcal{W}_t}^2,$$

其中常数 ν_t 和 μ_t 来自(3.2)。

证明。设 $\mathcal{P}_{\mathbb{T}}$ 为到 $\mathbb{T}$的正交投影算子，其中 $\mathcal{M}_r$ 是 $\boldsymbol{X}$在内积$\langle\cdot,\cdot\rangle_{\mathcal{W}_t}$ （定义见（2.10））下的切空间。那么，对于任意 $\boldsymbol{Z}\in\mathbb{R}^{n_1\times n_2}$，我们有

$$\widetilde{\mathcal{P}}_{\mathbb{T}}(\boldsymbol{Z})=\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{U}}^T\boldsymbol{L}_t^{\frac{1}{4}}\boldsymbol{Z}+\boldsymbol{Z}\boldsymbol{R}_t^{\frac{1}{4}}\widetilde{\boldsymbol{V}}\widetilde{\boldsymbol{V}}^T-\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{U}}^T\boldsymbol{L}_t^{\frac{1}{4}}\boldsymbol{Z}\boldsymbol{R}_t^{\frac{1}{4}}\widetilde{\boldsymbol{V}}\widetilde{\boldsymbol{V}}^T,\tag{3.6}$$

其中 $\boldsymbol{X}=\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$是 $\boldsymbol{X}$的紧奇异值分解，$\widetilde{\boldsymbol{U}}=\boldsymbol{U}(\boldsymbol{U}^T\boldsymbol{L}_t^{\frac{1}{4}}\boldsymbol{U})^{-\frac{1}{2}}$，且 $\widetilde{\boldsymbol{V}}=\boldsymbol{V}(\boldsymbol{V}^T\boldsymbol{R}_t^{\frac{1}{4}}\boldsymbol{V})^{-\frac{1}{2}}$。显然，$\widetilde{\mathcal{P}}_{\mathbb{T}}(\boldsymbol{X})=\boldsymbol{X}$。

使用与 [36, 引理 4。1], 类似的论证，我们得到

$$(\mathcal{I}-\widetilde{\mathcal{P}}_{\mathbb{T}_t})\boldsymbol{X}=(\widetilde{\mathcal{P}}_{\mathbb{T}}-\widetilde{\mathcal{P}}_{\mathbb{T}_t})\boldsymbol{X}=(\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{U}}^T\boldsymbol{L}_t^{\frac{1}{4}}-\widetilde{\boldsymbol{U}}_t\widetilde{\boldsymbol{U}}_t^T\boldsymbol{L}_t^{\frac{1}{4}})(\boldsymbol{X}-\boldsymbol{X}_t)(\boldsymbol{I}-\boldsymbol{R}_t^{\frac{1}{4}}\widetilde{\boldsymbol{V}}_t\widetilde{\boldsymbol{V}}_t^T).\tag{3.7}$$

这一结果结合引理 3.3、3.5 和3.6，可得

$$\begin{aligned}\|(\mathcal{I}-\widetilde{\mathcal{P}}_{\mathbb{T}_t})\boldsymbol{X}\|_{\mathcal{W}_t}&\le\mathrm{cond}(\boldsymbol{R}_t^{-\frac{1}{4}})\cdot\|(\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{U}}^T\boldsymbol{L}_t^{\frac{1}{4}}-\widetilde{\boldsymbol{U}}_t\widetilde{\boldsymbol{U}}_t^T\boldsymbol{L}_t^{\frac{1}{4}})\|_{\boldsymbol{L}_t^{\frac{1}{4}}}\|\boldsymbol{X}-\boldsymbol{X}_t\|_{\mathcal{W}_t}\|\boldsymbol{I}-\boldsymbol{R}_t^{\frac{1}{4}}\widetilde{\boldsymbol{V}}_t\widetilde{\boldsymbol{V}}_t^T\|_{\boldsymbol{R}_t^{\frac{1}{4}}}\\&\le\frac{\mathrm{cond}(\boldsymbol{R}_t^{-\frac{1}{4}})\cdot\mathrm{cond}(\boldsymbol{L}_t^{\frac{3}{8}})}{\sigma_{\min}(\boldsymbol{X})}\|\boldsymbol{X}_t-\boldsymbol{X}\|_2\|\boldsymbol{X}_t-\boldsymbol{X}\|_{\mathcal{W}_t}\\&\le\frac{\mathrm{cond}(\boldsymbol{R}_t^{-\frac{1}{4}})\cdot\mathrm{cond}(\boldsymbol{L}_t^{\frac{3}{8}})}{\sigma_{\min}(\boldsymbol{X})}\|\boldsymbol{X}_t-\boldsymbol{X}\|_F\|\boldsymbol{X}_t-\boldsymbol{X}\|_{\mathcal{W}_t}\\&\le\frac{\mathrm{cond}(\boldsymbol{R}_t^{-\frac{1}{4}})\cdot\mathrm{cond}(\boldsymbol{L}_t^{\frac{3}{8}})}{\nu_t^{\frac{1}{2}}\cdot\sigma_{\min}(\boldsymbol{X})}\|\boldsymbol{X}_t-\boldsymbol{X}\|_{\mathcal{W}_t}^2\le\frac{\mu_t^{\frac{5}{4}}}{\nu_t^{\frac{7}{4}}\cdot\sigma_{\min}(\boldsymbol{X})}\|\boldsymbol{X}-\boldsymbol{X}_t\|_{\mathcal{W}_t}^2.\end{aligned}$$

证明至此完成。 □

以下两个引理与 RIP 有关。

引理 3.8。设 $\boldsymbol{Z}_1$和 $\boldsymbol{Z}_2$是秩分别为 r_1和 r_2 的两个矩阵，且 $r_1+r_2\le$最小$\{n_1,n_2\}$ 并满足 $\langle\boldsymbol{Z}_1,\boldsymbol{Z}_2\rangle_{\mathcal{W}_t}=0$。假设 $\mathcal{A}$ 满足 *RIP* (3.1)，其常数为 $\delta_{r_1+r_2}$。则有

$$\left|\langle\mathcal{A}\boldsymbol{Z}_1,\mathcal{A}\boldsymbol{Z}_2\rangle\right|\le\frac{1}{2}\big((\nu_t^{-1}-\mu_t^{-1})+(\nu_t^{-1}+\mu_t^{-1})\delta_{r_1+r_2}\big)\|\boldsymbol{Z}_1\|_{\mathcal{W}_t}\|\boldsymbol{Z}_2\|_{\mathcal{W}_t},$$

其中 ν_t 和 μ_t 来自引理 3.2。

证明。 不失一般性，我们假设 $\|\boldsymbol{Z}_1\|_{\mathcal{W}_t} = \|\boldsymbol{Z}_2\|_{\mathcal{W}_t} = 1$。根据 RIP 和引理 3.2，我们有

$$\|\mathcal{A}(\boldsymbol{Z}_1 \pm \boldsymbol{Z}_2)\|_2^2 \geq (1-\delta_{r_1+r_2})\|\boldsymbol{Z}_1 \pm \boldsymbol{Z}_2\|_F^2 \geq \mu_t^{-1}(1-\delta_{r_1+r_2})\|\boldsymbol{Z}_1 \pm \boldsymbol{Z}_2\|_{\mathcal{W}_t}^2.$$

类似地，

$$\|\mathcal{A}(\boldsymbol{Z}_1 \pm \boldsymbol{Z}_2)\|_2^2 \leq (1+\delta_{r_1+r_2})\|\boldsymbol{Z}_1 \pm \boldsymbol{Z}_2\|_F^2 \leq \nu_t^{-1}(1+\delta_{r_1+r_2})\|\boldsymbol{Z}_1 \pm \boldsymbol{Z}_2\|_{\mathcal{W}_t}^2.$$

由于 $\langle \boldsymbol{Z}_1, \boldsymbol{Z}_2 \rangle_{\mathcal{W}_t} = 0$ 意味着 $\|\boldsymbol{Z}_1 \pm \boldsymbol{Z}_2\|_{\mathcal{W}_t}^2 = 2$，上述两个不等式意味着

$$2\mu_t^{-1}(1-\delta_{r_1+r_2}) \leq \|\mathcal{A}(\boldsymbol{Z}_1 \pm \boldsymbol{Z}_2)\|_2^2 \leq 2\nu_t^{-1}(1+\delta_{r_1+r_2}).$$

因此，

$$\begin{aligned}|\langle \mathcal{A}\boldsymbol{Z}_1, \mathcal{A}\boldsymbol{Z}_2\rangle| &= \frac{1}{4}\big|\|\mathcal{A}(\boldsymbol{Z}_1+\boldsymbol{Z}_2)\|_2^2 - \|\mathcal{A}(\boldsymbol{Z}_1-\boldsymbol{Z}_2)\|_2^2\big| \leq \frac{1}{4}\big|2\nu_t^{-1}(1+\delta_{r_1+r_2}) - 2\mu_t^{-1}(1-\delta_{r_1+r_2})\big| \\ &= \frac{1}{2}\left((\nu_t^{-1}-\mu_t^{-1}) + (\nu_t^{-1}+\mu_t^{-1})\delta_{r_1+r_2}\right).\end{aligned}$$

□

引理 3.9。 假设 $\mathcal{A}$ 满足 *RIP* (3.1)，其常数为 δ_{3r}。则

$$\|\widetilde{\mathcal{P}}_{\mathbb{T}_t}\mathcal{W}_t^{-1}\mathcal{A}^*\mathcal{A}(\mathcal{I}-\widetilde{\mathcal{P}}_{\mathbb{T}_t})\boldsymbol{X}\|_{\mathcal{W}_t} \leq \frac{1}{2}\left((\nu_t^{-1}-\mu_t^{-1}) + (\nu_t^{-1}+\mu_t^{-1})\delta_{3r}\right) \|(\mathcal{I}-\widetilde{\mathcal{P}}_{\mathbb{T}_t})\boldsymbol{X}\|_{\mathcal{W}_t}.$$

证明。 直接计算可得

$$\begin{aligned}\|\widetilde{\mathcal{P}}_{\mathbb{T}_t}\mathcal{W}_t^{-1}\mathcal{A}^*\mathcal{A}(\mathcal{I}-\widetilde{\mathcal{P}}_{\mathbb{T}_t})\boldsymbol{X}\|_{\mathcal{W}_t} &= \sup_{\|\boldsymbol{Z}\|_{\mathcal{W}_t}=1} \big|\langle \widetilde{\mathcal{P}}_{\mathbb{T}_t}\mathcal{W}_t^{-1}\mathcal{A}^*\mathcal{A}(\mathcal{I}-\widetilde{\mathcal{P}}_{\mathbb{T}_t})\boldsymbol{X}, \boldsymbol{Z}\rangle_{\mathcal{W}_t}\big| \\ &= \sup_{\|\boldsymbol{Z}\|_{\mathcal{W}_t}=1} |\langle \mathcal{A}(\mathcal{I}-\widetilde{\mathcal{P}}_{\mathbb{T}_t})\boldsymbol{X}, \mathcal{A}\widetilde{\mathcal{P}}_{\mathbb{T}_t}\boldsymbol{Z}\rangle|.\end{aligned}$$

~ ~现在我们将引理 3.8 应用于 $\boldsymbol{Z}_1 = (\mathcal{I}-\mathcal{P}_{\mathbb{T}_t})\boldsymbol{X}$ 以及 $\boldsymbol{Z}_2 = \mathcal{P}_{\mathbb{T}_t}\boldsymbol{Z}$。容易验证，秩$(\boldsymbol{Z}_1)$ $= r$，秩$(\boldsymbol{Z}_2)$ $= 2r$，且 $\langle \boldsymbol{Z}_1, \boldsymbol{Z}_2\rangle_{\mathcal{W}_t} = 0$。因此，引理 3.8 给出

$$\begin{aligned}\|\widetilde{\mathcal{P}}_{\mathbb{T}_t}\mathcal{W}_t^{-1}\mathcal{A}^*\mathcal{A}\big(\mathcal{I}-\widetilde{\mathcal{P}}_{\mathbb{T}_t}\big)\boldsymbol{X}\|_{\mathcal{W}_t} &\leq \sup_{\|\boldsymbol{Z}\|_{\mathcal{W}_t}=1} \frac{1}{2}\left((\nu_t^{-1}-\mu_t^{-1}) + (\nu_t^{-1}+\mu_t^{-1})\delta_{3r}\right)\|(\mathcal{I}-\widetilde{\mathcal{P}}_{\mathbb{T}_t})\boldsymbol{X}\|_{\mathcal{W}_t}\|\widetilde{\mathcal{P}}_{\mathbb{T}_t}\boldsymbol{Z}\|_{\mathcal{W}_t} \\ &\leq \frac{1}{2}\left((\nu_t^{-1}-\mu_t^{-1}) + (\nu_t^{-1}+\mu_t^{-1})\delta_{3r}\right)\|(\mathcal{I}-\widetilde{\mathcal{P}}_{\mathbb{T}_t})\boldsymbol{X}\|_{\mathcal{W}_t}.\end{aligned}$$

□

3.2 定理 3.2 的证明

现在我们给出主要结果定理 3.2 的证明。

定理 *3.2* 的证明。 由于 $\boldsymbol{X}_{t+1}$ 是 r 的最佳秩 $\boldsymbol{W}_t$ 近似（相对于 Frobenius 范数），利用 Frobenius 范数与 $\|\cdot\|_{\mathcal{W}_t}$ 范数之间的等价性，可得

$$\begin{aligned}\|\boldsymbol{X}_{t+1}-\boldsymbol{X}\|_{\mathcal{W}_t} &\leq \|\boldsymbol{X}_{t+1}-\boldsymbol{W}_t\|_{\mathcal{W}_t} + \|\boldsymbol{W}_t-\boldsymbol{X}\|_{\mathcal{W}_t} \leq \sqrt{\mu_t}\|\boldsymbol{X}_{t+1}-\boldsymbol{W}_t\|_F + \|\boldsymbol{W}_t-\boldsymbol{X}\|_{\mathcal{W}_t} \\ &\leq (1+\sqrt{\mu_t/\nu_t})\|\boldsymbol{W}_t-\boldsymbol{X}\|_{\mathcal{W}_t} := (1+\sqrt{\rho_t})\|\boldsymbol{W}_t-\boldsymbol{X}\|_{\mathcal{W}_t}.\end{aligned}$$

在此处及下文中，我们使用 $\rho_t = \frac{\mu_t}{\nu_t}$。因此，只需估计 $\|\boldsymbol{W}_t-\boldsymbol{X}\|_{\mathcal{W}_t}$。利用算法 1 以及 $\mathcal{W}_t$的定义，我们可以将 $\boldsymbol{W}_t$ 表示为

$$\boldsymbol{W}_t = \boldsymbol{X}_t - \alpha_t\widetilde{\mathcal{P}}_{\mathbb{T}_t}\big(\boldsymbol{L}_t^{-\frac{1}{4}}\boldsymbol{G}_t\boldsymbol{R}_t^{-\frac{1}{4}}\big) = \boldsymbol{X}_t - \alpha_t\widetilde{\mathcal{P}}_{\mathbb{T}_t}\mathcal{W}_t^{-1}\boldsymbol{G}_t.$$

根据 $\boldsymbol{G}_t$ 的定义，我们得到

$$\begin{aligned}\|\boldsymbol{W}_t-\boldsymbol{X}\|_{\mathcal{W}_t} &= \|\boldsymbol{X}_t-\alpha_t\widetilde{\mathcal{P}}_{\mathbb{T}_t}\mathcal{W}_t^{-1}\boldsymbol{G}_t-\boldsymbol{X}\|_{\mathcal{W}_t} = \|\boldsymbol{X}_t-\boldsymbol{X}-\alpha_t\widetilde{\mathcal{P}}_{\mathbb{T}_t}\mathcal{W}_t^{-1}\mathcal{A}^*\mathcal{A}(\boldsymbol{X}_t-\boldsymbol{X})\|_{\mathcal{W}_t}\\ &\le \|(\mathcal{I}-\alpha_t\widetilde{\mathcal{P}}_{\mathbb{T}_t}\mathcal{W}_t^{-1}\mathcal{A}^*\mathcal{A}\widetilde{\mathcal{P}}_{\mathbb{T}_t})(\boldsymbol{X}_t-\boldsymbol{X})\|_{\mathcal{W}_t}+\alpha_t\|\widetilde{\mathcal{P}}_{\mathbb{T}_t}\mathcal{W}_t^{-1}\mathcal{A}^*\mathcal{A}(\mathcal{I}-\widetilde{\mathcal{P}}_{\mathbb{T}_t})(\boldsymbol{X}_t-\boldsymbol{X})\|_{\mathcal{W}_t}\\ &\le \underbrace{\|(\widetilde{\mathcal{P}}_{\mathbb{T}_t}-\alpha_t\widetilde{\mathcal{P}}_{\mathbb{T}_t}\mathcal{W}_t^{-1}\mathcal{A}^*\mathcal{A}\widetilde{\mathcal{P}}_{\mathbb{T}_t})(\boldsymbol{X}_t-\boldsymbol{X})\|_{\mathcal{W}_t}}_{I_1}+\underbrace{\|(\mathcal{I}-\widetilde{\mathcal{P}}_{\mathbb{T}_t})(\boldsymbol{X}_t-\boldsymbol{X})\|_{\mathcal{W}_t}}_{I_2}\\ &\quad+\alpha_t\underbrace{\|\widetilde{\mathcal{P}}_{\mathbb{T}_t}\mathcal{W}_t^{-1}\mathcal{A}^*\mathcal{A}(\mathcal{I}-\widetilde{\mathcal{P}}_{\mathbb{T}_t})(\boldsymbol{X}_t-\boldsymbol{X})\|_{\mathcal{W}_t}}_{I_3}.\end{aligned}$$

选择

$$\alpha_t=\frac{2}{\mu_t^{-1}+\nu_t^{-1}}. \tag{3.8}$$

我们可以分别估计这三项如下：

- 估计 I_1。由于算子 $\widetilde{\mathcal{P}}_{\mathbb{T}_t}-\alpha_t\widetilde{\mathcal{P}}_{\mathbb{T}_t}\mathcal{W}_t^{-1}\mathcal{A}^*\mathcal{A}\widetilde{\mathcal{P}}_{\mathbb{T}_t}$ 在 $\langle\cdot,\cdot\rangle_{\mathcal{W}_t}$下是自伴的，因此我们有

$$\begin{aligned}\|\widetilde{\mathcal{P}}_{\mathbb{T}_t}-\alpha_t\widetilde{\mathcal{P}}_{\mathbb{T}_t}\mathcal{W}_t^{-1}\mathcal{A}^*\mathcal{A}\widetilde{\mathcal{P}}_{\mathbb{T}_t}\|_{\mathcal{W}_t} &= \sup_{\|\boldsymbol{Z}\|_{\mathcal{W}_t}=1}|\langle(\widetilde{\mathcal{P}}_{\mathbb{T}_t}-\alpha_t\widetilde{\mathcal{P}}_{\mathbb{T}_t}\mathcal{W}_t^{-1}\mathcal{A}^*\mathcal{A}\widetilde{\mathcal{P}}_{\mathbb{T}_t})\boldsymbol{Z},\boldsymbol{Z}\rangle_{\mathcal{W}_t}|\\ &= \sup_{\|\boldsymbol{Z}\|_{\mathcal{W}_t}=1}\big|\|\widetilde{\mathcal{P}}_{\mathbb{T}_t}\boldsymbol{Z}\|_{\mathcal{W}_t}^2-\alpha_t\langle\mathcal{A}^*\mathcal{A}\widetilde{\mathcal{P}}_{\mathbb{T}_t}\boldsymbol{Z},\mathcal{W}_t^{-1}\widetilde{\mathcal{P}}_{\mathbb{T}_t}\boldsymbol{Z}\rangle_{\mathcal{W}_t}\big|\\ &= \sup_{\|\boldsymbol{Z}\|_{\mathcal{W}_t}=1}\big|\|\widetilde{\mathcal{P}}_{\mathbb{T}_t}\boldsymbol{Z}\|_{\mathcal{W}_t}^2-\alpha_t\|\mathcal{A}\widetilde{\mathcal{P}}_{\mathbb{T}_t}\boldsymbol{Z}\|_2^2\big|.\end{aligned} \tag{3.9}$$

  由于秩$(\mathcal{P}_{\mathbb{T}_t}\boldsymbol{Z})\le 2r$，RIP (3.1) 和引理 3.3 可得

$$\begin{aligned}\mu_t^{-1}(1-\delta_{2r})\|\widetilde{\mathcal{P}}_{\mathbb{T}_t}\boldsymbol{Z}\|_{\mathcal{W}_t}^2\le(1-\delta_{2r})\|\widetilde{\mathcal{P}}_{\mathbb{T}_t}\boldsymbol{Z}\|_F^2\le\|\mathcal{A}\widetilde{\mathcal{P}}_{\mathbb{T}_t}\boldsymbol{Z}\|_2^2\\ \le(1+\delta_{2r})\|\widetilde{\mathcal{P}}_{\mathbb{T}_t}\boldsymbol{Z}\|_F^2\le\nu_t^{-1}(1+\delta_{2r})\|\widetilde{\mathcal{P}}_{\mathbb{T}_t}\boldsymbol{Z}\|_{\mathcal{W}_t}^2.\end{aligned} \tag{3.10}$$

  结合（3.9）与（3.10），我们得到

$$\begin{aligned}\|\widetilde{\mathcal{P}}_{\mathbb{T}_t}-\alpha_t\widetilde{\mathcal{P}}_{\mathbb{T}_t}\mathcal{W}_t^{-1}\mathcal{A}^*\mathcal{A}\widetilde{\mathcal{P}}_{\mathbb{T}_t}\|_{\mathcal{W}_t} &\le \max\left\{\left|1-\alpha_t\nu_t^{-1}(1+\delta_{2r})\right|,\left|1-\alpha_t\mu_t^{-1}(1-\delta_{2r})\right|\right\}\sup_{\|\boldsymbol{Z}\|_{\mathcal{W}_t}=1}\|\widetilde{\mathcal{P}}_{\mathbb{T}_t}\boldsymbol{Z}\|_{\mathcal{W}_t}^2\\ &\le \max\left\{\left|1-\alpha_t\nu_t^{-1}(1+\delta_{2r})\right|,\left|1-\alpha_t\mu_t^{-1}(1-\delta_{2r})\right|\right\}\\ &= 1-\frac{2}{1+\rho_t}(1-\delta_{2r}),\end{aligned} \tag{3.11}$$

  其中，在最后一个等式中，我们使用了（3.8）。

- 的估计 I_2。由于 $(\mathcal{I}-\mathcal{P}_{\mathbb{T}_t})\widetilde{\boldsymbol{X}_t}=\boldsymbol{0}$，引理3.7 表明

$$\|(\mathcal{I}-\widetilde{\mathcal{P}}_{\mathbb{T}_t})(\boldsymbol{X}_t-\boldsymbol{X})\|_{\mathcal{W}_t}\le\frac{\rho_t^{\frac{5}{4}}}{\nu_t^{\frac{1}{2}}\sigma_{\min}(\boldsymbol{X})}\|\boldsymbol{X}_t-\boldsymbol{X}\|_{\mathcal{W}_t}^2. \tag{3.12}$$

- 对 I_3 的估计。最后，我们推导出 $\alpha_t\|\widetilde{\mathcal{P}}_{\mathbb{T}_t}\mathcal{W}_t^{-1}\mathcal{A}^*\mathcal{A}(\mathcal{I}-\widetilde{\mathcal{P}}_{\mathbb{T}_t})(\boldsymbol{X}_t-\boldsymbol{X})\|_{\mathcal{W}_t}$ 的一个界。利用引理3.9和方程（3.8），可得

$$\alpha_t\|\widetilde{\mathcal{P}}_{\mathbb{T}_t}\mathcal{W}_t^{-1}\mathcal{A}^*\mathcal{A}(\mathcal{I}-\widetilde{\mathcal{P}}_{\mathbb{T}_t})\boldsymbol{X}\|_{\mathcal{W}_t}\le\left(\frac{\rho_t-1}{\rho_t+1}+\delta_{3r}\right)\|\boldsymbol{X}_t-\boldsymbol{X}\|_{\mathcal{W}_t}. \tag{3.13}$$

结合（3.11）、（3.12）和（3.13），可得

$$\begin{aligned}
&\|\boldsymbol{X}_{t+1}-\boldsymbol{X}\|_{\mathcal{W}_t} \leq (1+\sqrt{\rho_t})\|\boldsymbol{W}_t-\boldsymbol{X}\|_{\mathcal{W}_t} \\
&\leq (1+\sqrt{\rho_t})\left(1-\frac{2}{1+\rho_t}(1-\delta_{2r})+\frac{\rho_t-1}{\rho_t+1}+\delta_{3r}+\frac{\rho_t^{\frac{5}{4}}}{\nu_t^{\frac{1}{2}}\sigma_{\min}(\boldsymbol{X})}\|\boldsymbol{X}_t-\boldsymbol{X}\|_{\mathcal{W}_t}\right)\|\boldsymbol{X}_t-\boldsymbol{X}\|_{\mathcal{W}_t} \\
&\leq (1+\sqrt{\rho_t})\left(1-\frac{2}{1+\rho_t}(1-\delta_{2r})+\frac{\rho_t-1}{\rho_t+1}+\delta_{3r}+\frac{\rho_t^{\frac{7}{4}}}{\sigma_{\min}(\boldsymbol{X})}\|\boldsymbol{X}_t-\boldsymbol{X}\|_F\right)\|\boldsymbol{X}_t-\boldsymbol{X}\|_{\mathcal{W}_t}.
\end{aligned}$$

根据引理 3.3，我们进一步有

$$\begin{aligned}
&\|\boldsymbol{X}_{t+1}-\boldsymbol{X}\|_F \\
&\leq (\rho_t+\sqrt{\rho_t})\left(1-\frac{2}{1+\rho_t}(1-\delta_{2r})+\frac{\rho_t-1}{\rho_t+1}+\delta_{3r}+\frac{\rho_t^{\frac{7}{4}}}{\sigma_{\min}(\boldsymbol{X})}\|\boldsymbol{X}_t-\boldsymbol{X}\|_F\right)\|\boldsymbol{X}_t-\boldsymbol{X}\|_F.
\end{aligned} \tag{3.14}$$

现在令 $\epsilon_t=\|\boldsymbol{G}_t\|_{\vee}^2$ 并使用引理 3.3 得到 $\rho_t=\sqrt{2}$。将其代入（3.14），并利用事实 $\delta_{2r}\leq\delta_{3r}$，我们可以将（3.14）简化为

$$\|\boldsymbol{X}_{t+1}-\boldsymbol{X}\|_F \leq \left(0.895+5\delta_{3r}+\frac{2^{\frac{7}{8}}}{\sigma_{\min}(\boldsymbol{X})}\|\boldsymbol{X}_t-\boldsymbol{X}\|_F\right)\|\boldsymbol{X}_t-\boldsymbol{X}\|_F. \tag{3.15}$$

在算法 1中，我们将初始猜测设为 $\boldsymbol{X}_0=\mathcal{H}_r(\mathcal{A}^*(\boldsymbol{y}))$。根据 [36], 中的分析，我们有

$$\|\boldsymbol{X}_0-\boldsymbol{X}\|_F \leq 2\delta_{2r}\|\boldsymbol{X}\|_F \leq 2\delta_{3r}\sqrt{r}\sigma_{\max}(\boldsymbol{X}). \tag{3.16}$$

记

$$\mu=0.895+5\delta_{3r}+10\delta_{3r}\kappa\sqrt{r}.$$

如果 $\mu<1$，则我们使用归纳法证明

$$\|\boldsymbol{X}_{k+1}-\boldsymbol{X}\|_F \leq \mu\|\boldsymbol{X}_k-\boldsymbol{X}\|_F, \tag{3.17}$$

对所有 k。结合（3.15）和（3.16），我们得到（3.17）对 $k=0$成立。假设（3.17）对 k =0、1、$\ldots t-1$成立。那么我们有

$$\|\boldsymbol{X}_t-\boldsymbol{X}\|_F \leq \mu^t\|\boldsymbol{X}_0-\boldsymbol{X}\|_F \leq 2\delta_{3r}\sqrt{r}\sigma_{\max}(\boldsymbol{X}).$$

将其代入（3.15），得到（3.17）对于 $k=t$。因此，关于 k 的归纳法表明（3.17）对于任意 k均成立。此外，如果

$$\delta_{3r} \leq \frac{1}{50+100\sqrt{r}\kappa},$$

于是得到 $\mu\leq 0.995$。证毕。 □

4 数值实验

在本节中，我们将采用算法 1中描述的 PRGD 算法，求解低秩矩阵恢复问题，例如低秩矩阵补全、低秩矩阵感知和相位恢复。我们将 PRGD 的性能与其他最先进算法进行比较，包括归一化迭代硬阈值（NIHT）[32], 采用典范度量的黎曼梯度下降（RGD）[7, 26, 29,35, 36, 47], 以及采用 Shampoo 预条件器的 RGD [49]，其定义见（2.7）。实验结果表明，与其他方法相比，PRGD 效率很高。所有实验均使用 MATLAB R2020b，在配备 2.4GHz i5-10600T CPU 和 16GB 内存的台式计算机上进行。实验中涉及的奇异值分解（SVD）使用 MATLAB R2020b 中的 PROPACK 完成。

4.1 低秩矩阵补全

我们首先通过评估 PRGD 求解低秩矩阵补全问题的效率，考察其性能。在该问题中，（1.2）中的测量矩阵具有如下形式： $\boldsymbol{A}_i = \boldsymbol{e}_{r_i}\boldsymbol{e}_{c_i}^T$ 对所有（r_i， c_i）$\in \Omega$，其中 Ω 是 $[n_1] \times [n_2]$ 的一个子集，且满足 $|\Omega| = m$。矩阵补全的目标是根据低秩矩阵 $\boldsymbol{X}$ 在 Ω 上部分观测到的条目恢复该矩阵。

4.1.1 模拟数据

我们在模拟数据上测试 PRGD 算法。为生成大小为 $n_1 \times n_2$、秩为 r 的矩阵 $\boldsymbol{X}$，我们从 $[0,1]$ 中的均匀分布采样 $\boldsymbol{X}_L \in \mathbb{R}^{n_1 \times r}$ 和 $\boldsymbol{X}_R \in \mathbb{R}^{n_2 \times r}$ 的元素。然后设定 $\boldsymbol{X} = \boldsymbol{X}_L\boldsymbol{X}_R^T$。观测条目集合 Ω 从所有基数为 m 的子集中均匀随机采样。我们将采样率 p 定义为 $p = \frac{m}{n_1 n_2}$，过采样率定义为

$$q = \frac{(n_1 + n_2 - r)r}{m}. \tag{4.1}$$

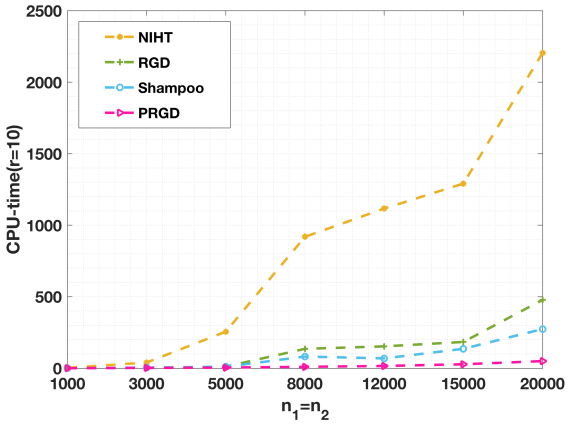
给定固定参数 (n_1, n_2, m, r)，我们得到固定的 p 和 q。对于每项实验，我们使用相同的 (n_1, n_2, m, r) 进行 5 次随机测试，并报告平均结果以供比较。

对矩阵规模的敏感性。 我们首先评估 PRGD 对矩阵规模变化的敏感性。为此，我们针对不同的矩阵规模 (n_1, n_2) 以及 r 和 q 的不同组合进行测试。我们比较不同算法获得满足 $\|\boldsymbol{X} - \boldsymbol{X}_t\|_F \le 10^{-4} \cdot \|\boldsymbol{X}\|_F$ 的 $\boldsymbol{X}_t$ 所需的 CPU 时间（秒）和迭代次数。

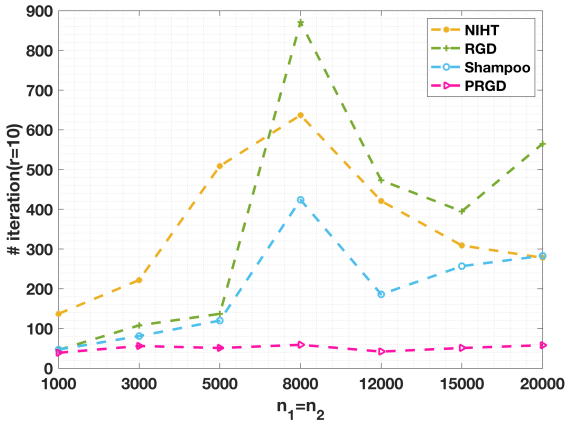
我们在图1中展示了结果。图1 表明，与其他所有算法相比，PRGD通常所需的迭代次数显著更少。此外，随着矩阵规模增大，其他三种算法所需的迭代次数明显增加，而PRGD的迭代次数仅略有增加。此外，正如第2.2.3节的分析所示，PRGD每次迭代所需的计算时间几乎与RGD和Shampoo相同，但远少于NIHT。因此，PRGD的总计算时间远少于其他三种算法。事实上，从图1 中还可以看出，随着矩阵规模增大，其他三种算法所需的计算时间显著增加。然而，PRGD所需的计算时间随矩阵规模增大仅缓慢上升。值得注意的是，当矩阵规模为20000年时，PRGD的计算时间显著低于其他三种算法。具体而言，PRGD所需的计算时间仅为NIHT、RGD和Shampoo的约 $1/20$、 $1/10$和 $1/5$ 倍。所有这些结果都表明，选择合适的度量是加速RGD的有效方法，而PRGD中的数据驱动度量远优于Shampoo所使用的度量。

对过采样率的敏感性。 我们研究 PRGD 对过采样率变化的敏感性 q。根据式（4.1）的定义，过采样率 q 是采样条目数与一个 $n_1 \times n_2$ 秩为 r 的矩阵的“真实维数”之比。过采样率越小，矩阵补全问题就越具挑战性，因为可用于估计未知条目的观测条目更少。我们对大小为 10000×10000、秩为 10 的矩阵进行测试，并将过采样率在集合 $\{1/3, 1/4, 1/5, 1/6, 1/7, 1/8, 1/9, 1/10, 1/11, 1/12, 1/13, 1/14, 1/15\}$中变化。寻找满足 $\boldsymbol{X}_t$ 的 $\|\boldsymbol{X} - \boldsymbol{X}_t\|_F \le 10^{-4} \cdot \|\boldsymbol{X}\|_F$所需的迭代次数和计算时间的结果见图2。

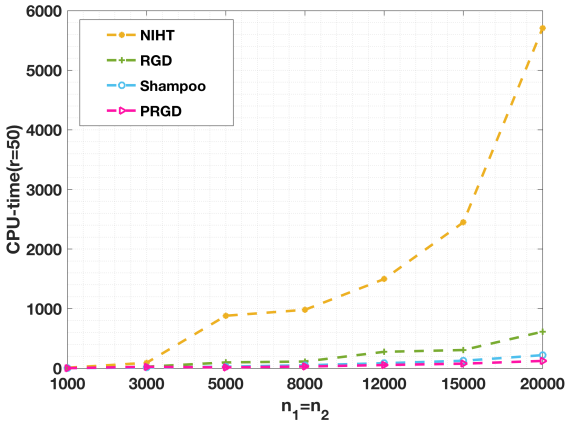
图中的结果2 表明，在所有情况下，PRGD 算法在计算时间和所需迭代次数方面均优于其他所有算法。此外，随着过采样率 q 增加，即问题变得更具挑战性，PRGD 在迭代次数和计算时间方面的优势也愈发明显。例如，当过采样率 q 设为 $1/3$时，这表明矩阵补全问题极具挑战性，而 PRGD 算法所需的迭代次数仅约为 $1/10$，所需的计算时间仅约为 $1/8$。



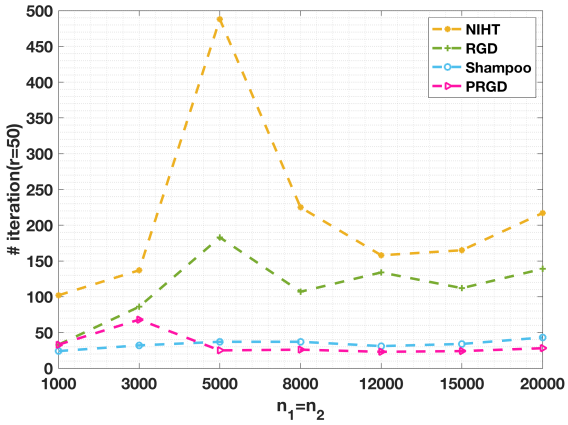
(a) CPU-time(s) $r = 10$ and $q = 5$



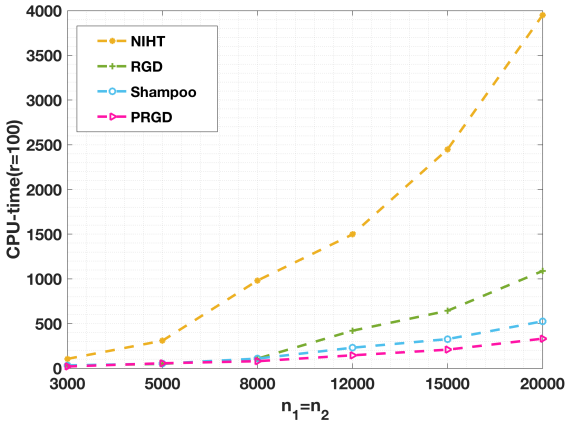
(b) ♯iteration $r = 10$ and $q = 5$



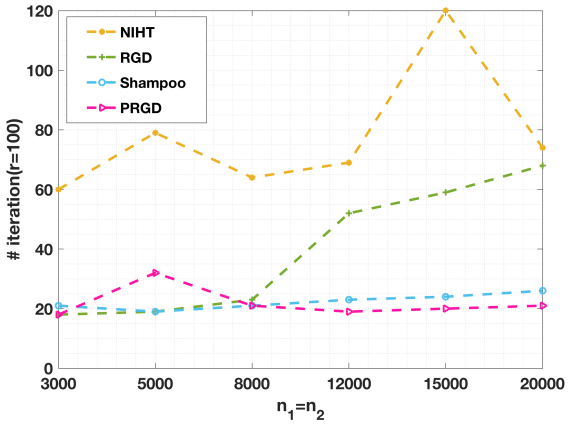
(c) CPU-time(s) $r = 50$ and $q = 6$



(d) ♯iteration $r = 50$ and $q = 6$



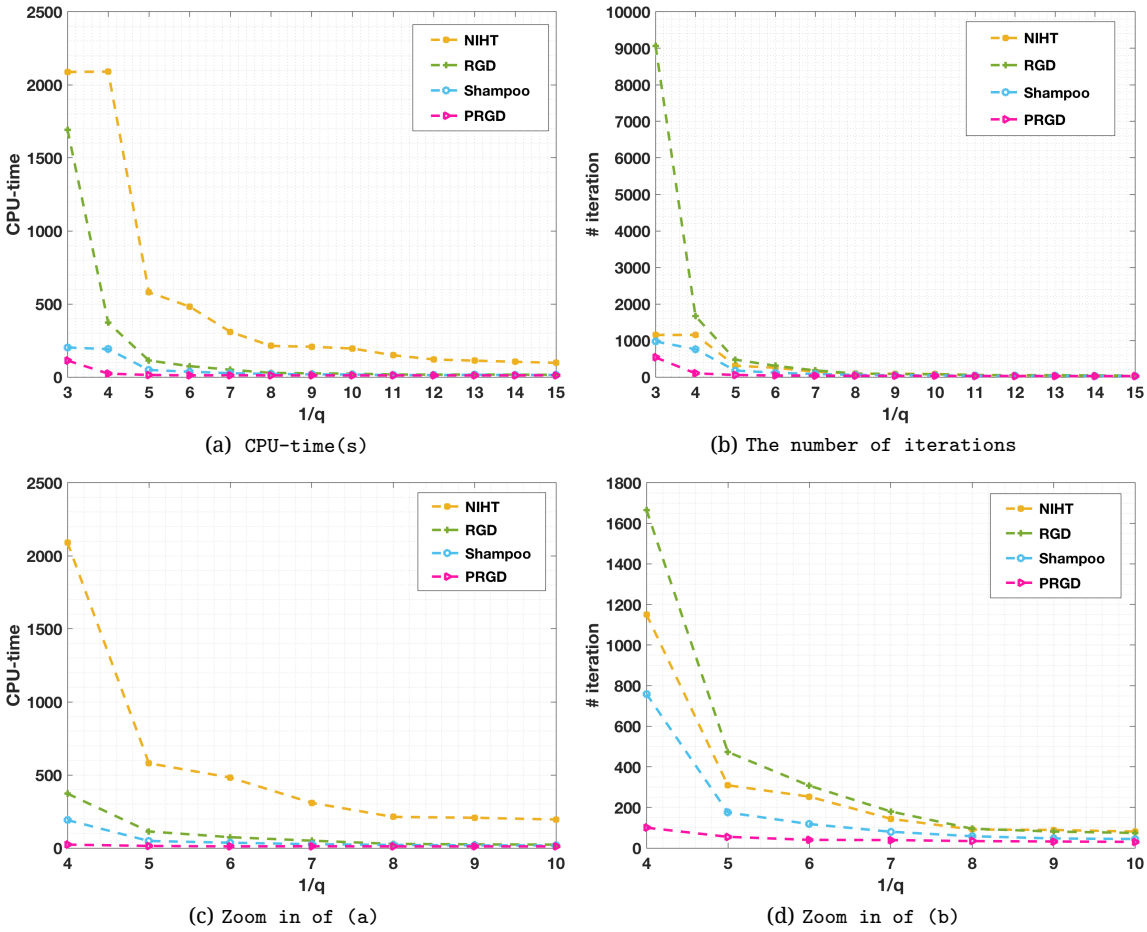
(e) CPU-time(s) $r = 100$ and $q = 7$



(f) ♯iteration $r = 100$ and $q = 7$

图1：模拟数据上矩阵补全的 CPU 时间（以秒为单位）和迭代次数结果。

相较于 RGD。这些观察结果进一步证实，我们构建的 PRGD 算法效率很高，且优于 RGD、Shampoo 和 NIHT。



(a) CPU-time(s)　(b) The number of iterations

(c) Zoom in of (a)　(d) Zoom in of (b)

图 2：模拟数据上矩阵补全的 CPU 时间（秒）和迭代次数。未知矩阵的大小为 10000×10000，秩为 $r = 10$。

对加性噪声的鲁棒性。为了评估算法对加性噪声的鲁棒性，我们使用向量向采样条目添加不同的噪声

$$e = \sigma \cdot \|\mathcal{P}_\Omega(\boldsymbol{X})\|_F \cdot \frac{\boldsymbol{w}}{\|\boldsymbol{w}\|_2}, \tag{4.2}$$

其中， $\boldsymbol{w}$ 的各个条目服从 $i.i.d$ 高斯随机变量，且 σ 称为噪声水平。实际上， σ也是观测数据的相对误差。我们对 σ 从 1×10^{-4} 到1×10^{-1}的 8 个不同取值进行了测试。我们测试秩为 10000×10000 的 $r = 10$ 矩阵，并固定过采样率 $q = 5$。当 $\|\boldsymbol{X}_{t+1} - \boldsymbol{X}_t\|_F \leq 10^{-5} \cdot \max\{1, \|\boldsymbol{X}_t\|_F\}$时，我们停止所有算法。结果见图3 和 4。图3 表明，所有算法都能抵抗加性噪声，因为恢复矩阵的误差仅与噪声水平成正比。此外，从图4可以看出，PRGD 的速度快于其他三种算法。噪声水平越高，这种效率优势就越明显。例如，当 $\sigma = 0.1$时，PRGD 算法所需的迭代次数仅约为 RGD 的 $1/10$，所需的计算时间仅约为 RGD 的$1/10$。

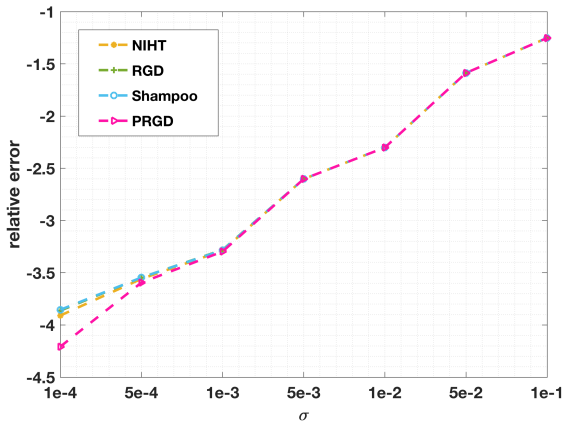
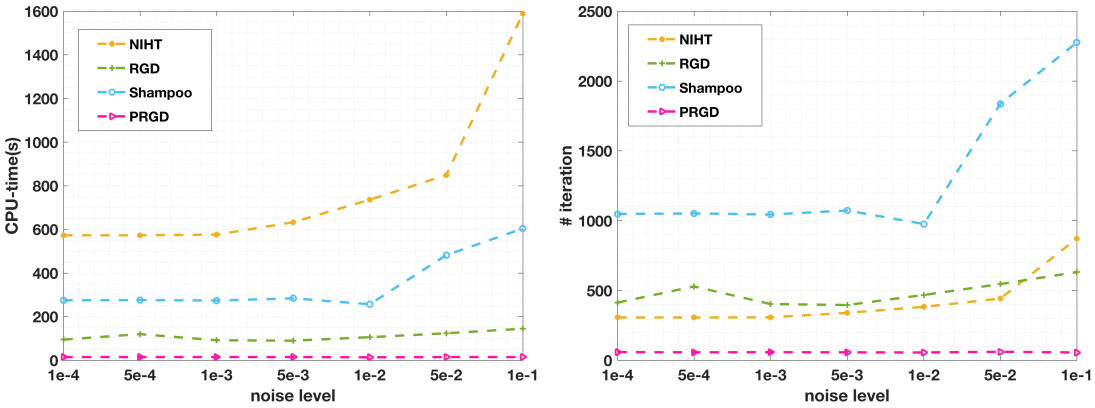
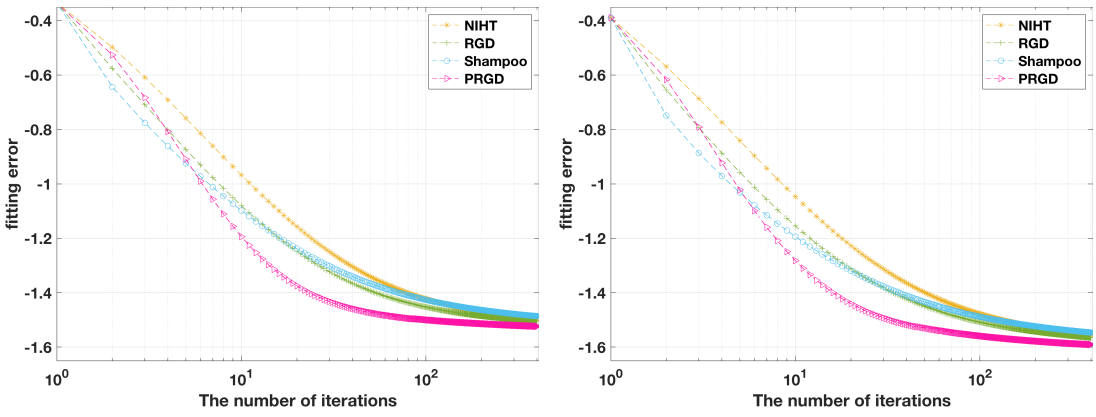


图 3：含噪声模拟数据上的矩阵补全恢复矩阵相对误差结果。未知矩阵的大小为 10000×10000，秩为 $r = 10$。



(a) CPU-time(s)　　(b) The number of iterations

图 4：在带噪声的模拟数据上进行矩阵补全的 CPU 时间（秒）和迭代次数结果。未知矩阵的大小为 10000×10000，秩为 $r = 10$。



(a) r=5　　(b) r=10

图5：真实数据上的矩阵补全拟合误差结果。我们分别将未知矩阵的秩设为 $r = 5$ 和 $r = 10$。

4.1.2 真实数据

我们在 Movielens 数据集上评估 PRGD 算法的性能。该数据集是真实数据，是矩阵补全问题中常用的数据集。具体而言，我们使用 Movielens 1M 数据集 [20], ，其中包含 2000年加入 Movielens 的 6,040 名用户对 3,952 部电影的匿名评分。由于该真实数据集的真实值及其秩未知，因此我们分别设定 $r = 5$ 和 $r = 10$，并比较拟合误差

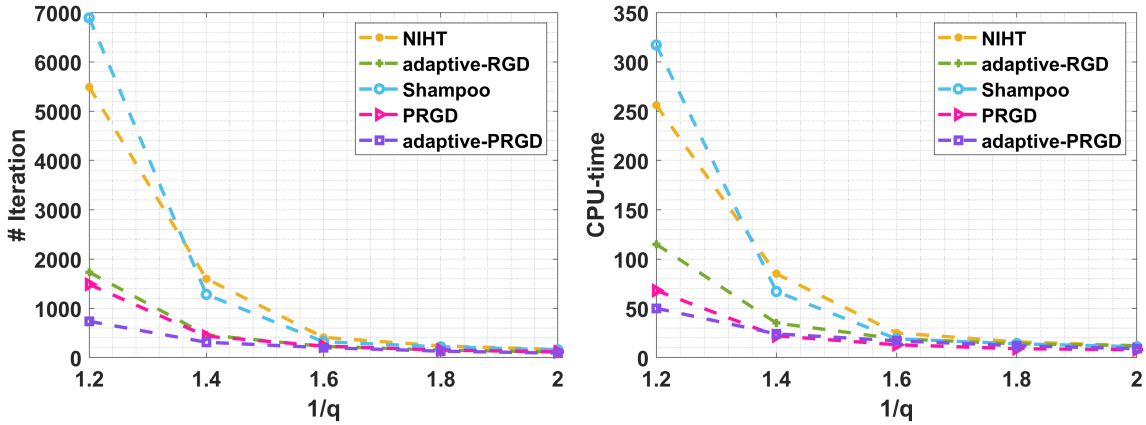
$$\frac{\|\mathcal{P}_\Omega(\boldsymbol{X} - \boldsymbol{X}_t)\|_F}{\|\mathcal{P}_\Omega(\boldsymbol{X})\|_F}$$

通过以相同的迭代次数运行所有算法。我们在图5中绘制拟合误差随迭代次数的变化曲线。从图中可以看出，PRGD 是使用低秩矩阵拟合 Movielens 数据集时效率最高的算法。具体而言，对于所有 $\boldsymbol{X}_t$ ，PRGD 的输出具有所有算法中最小的拟合误差 t，且达到相同拟合误差所需的迭代次数最少。值得注意的是，要将拟合误差降低至 $10^{-1.4}$，PRGD 所需的迭代次数仅为 RGD 的一半。

4.2 低秩矩阵感知

在本节中，我们进一步评估 PRGD 在矩阵感知问题中恢复低秩矩阵的能力，其中线性算子 $\boldsymbol{A}_i$、$i = 1$、2、.....、m、 $\mathcal{A}$（定义见（1.2））中的测量矩阵均为一般矩阵。在本实验中，我们使用独立同分布标准高斯随机变量生成 $\boldsymbol{A}_i$ 的元素。基础矩阵 $\boldsymbol{X}$ 构造为 $\boldsymbol{X} = \boldsymbol{X_L}\boldsymbol{X_R}^T$，其中 $\boldsymbol{X_L} \in \mathbb{R}^{n_1 \times r}$ 和 $\boldsymbol{X_R} \in \mathbb{R}^{n_2 \times r}$ 的元素均为独立同分布标准高斯随机变量。测量向量 $\boldsymbol{y} \in \mathbb{R}^m$ 生成如下： $\boldsymbol{y} = \mathcal{A}\boldsymbol{X}$。我们将采样率定义为 $p = \frac{m}{n_1 n_2}$ ，并将过采样率 q 定义为（4.1）中所定义的形式。

在本实验中，采用固定步长的 RGD 表现不佳，并且需要许多迭代次数。因此，我们不纳入 RGD 的比较，而是将采用自适应最速下降步长的 RGD（称为 adaptive-RGD）与采用自适应最速下降步长的 PRGD（称为 adaptive-PRGD）进行比较。



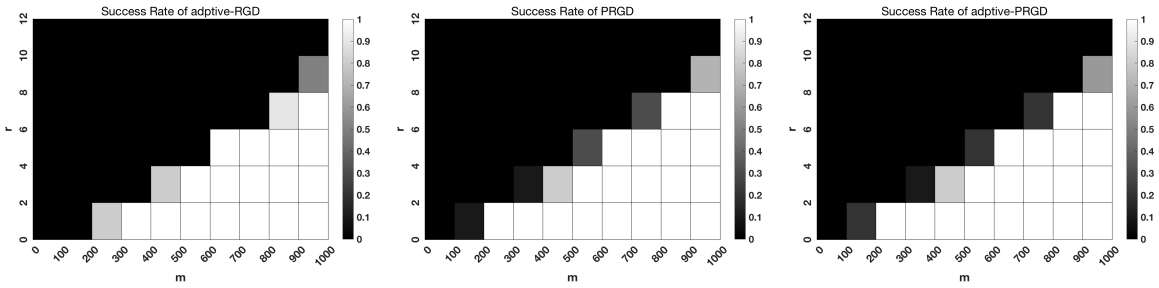
(a) The number of iterations　　(b) CPU-time(s)

图6：模拟数据上矩阵感知的 CPU 时间（以秒为单位）和迭代次数结果。未知矩阵的大小为 150×150，秩为 $r = 10$。

对过采样率的敏感性。 我们研究 PRGD 对过采样率 q 变化的敏感性，其中 q 的值越大，意味着问题越具挑战性。我们固定测量矩阵的大小为 $n_1 = n_2 = 150$ ，其秩为 $r = 10$，同时改变过采样率

$1/q \in \{1.2, 1.4, 1.6, 1.8, 2.0\}$。获得满足 $\boldsymbol{X}_t$ 的迭代次数和计算时间结果 $\|\boldsymbol{X} - \boldsymbol{X}_t\|_F \leq 10^{-4} \cdot \|\boldsymbol{X}\|_F$如图 6所示。

从图6(a) 可以看出，在迭代次数方面，adaptive-PRGD 优于其他四种算法。PRGD 的迭代次数与 adaptive-RGD 相近。具体而言，当过采样率 $1/q = 1.2$时，adaptive-PRGD 的迭代次数约为 adaptive-RGD、NIHT、Shampoo 和 PRGD 的 $1/2.35$， $1/7.5$，$1/9.4$和 $1/2$ 。从图6(b) 可以看出，adaptive-PRGD 的计算时间始终少于 adaptive-RGD、Shampoo、NIHT 和 PRGD。当过采样率 $1/q = 1.2$ 时，adaptive-PRGD 的计算时间约为 $1/2.3$， $1/5.1$， $1/6.3$和 $1/1.3$ 分别为 adaptive-RGD、NIHT、Shampoo 和 PRGD 的计算时间。PRGD 的计算时间少于 adaptive-RGD。这是因为，尽管 adaptive-RGD 和 PRGD 的迭代次数相近，adaptive-RGD 还需要额外花费时间计算最速下降步长。



(a) adaptive-RGD (b) PRGD (c) adaptive-PRGD

图 7：模拟数据上矩阵感知的成功率结果。未知矩阵的大小为 50×50。成功率通过方框的灰度表示，其中白色表示成功率为 100%，黑色表示成功率为 0%。

成功率。我们测试 PRGD 算法的恢复能力。我们比较 adaptive-PRGD、adaptive-RGD 和 PRGD 在恢复测试矩阵 $\boldsymbol{X}$ 时的成功率。如果输出值 $\boldsymbol{X}_t$ 满足以下条件，则认为该算法成功恢复了秩为 r 的测试矩阵 $\boldsymbol{X}$ 。

$$\frac{\|\boldsymbol{X}_t - \boldsymbol{X}\|_F}{\|\boldsymbol{X}\|_F} \leq 10^{-4}$$

在 500 次迭代以内。我们固定矩阵大小为 $n_1 = n_2 = 50$，并改变秩 $r \in \{2, 4, 6, 8, 10, 12\}$ 和测量次数 $m \in \{100, 200, 300, 400, 500, 600, 700, 800, 900, 1000\}$。我们在图7中展示结果，其中所有算法的成功率均通过 10 次独立试验得出。

图 7 中的结果表明，adaptive-PRGD 的成功率与 PRGD 相当，且优于 adaptive-RGD。当秩 r 和测量次数 m 较小时，adaptive-PRGD 的成功率高于 adaptive-RGD。例如，当 $r = 2$ 且 $m = 300$时，PRGD 和 adaptive-PRGD 均能以 100% 的成功率恢复，而 adaptive-RGD 的成功率为 80%。当秩 r 相对较大时，adaptive-RGD 需要更多测量次数才能达到 100% 的成功率。例如，当 $r = 8$时，若 $m = 900$，adaptive-PRGD 和 PRGD 均能以 100% 的成功率恢复，而 adaptive-RGD 的成功率为 90%。当 $m = 800$ 时，adaptive-PRGD 和 PRGD 的成功率分别为 30% 和 20%，而 adaptive-RGD 的成功率为 0%。

4.3 相位恢复

最后，我们评估 PRGD 在相位恢复问题上的性能。为简单起见，我们考虑实数情形，但所有方法都可应用于复数情形。在相位恢复中，目标是求得一个

满足以下无相位方程组的向量 $\boldsymbol{x} \in \mathbb{R}^n$：

$$|\boldsymbol{A}x|^2 = \boldsymbol{y}, \tag{4.3}$$

其中， $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ 和 $\boldsymbol{y} \in \mathbb{R}^m$ 已知。我们将 $\boldsymbol{A}$的第 i行记为 $\boldsymbol{a}_i^T$。设 $\mathcal{A}$ 为一个线性算子，它将一个 $n \times n$ 矩阵映射为长度为 m的向量，定义如下：

$$\mathcal{A}\boldsymbol{Z} = \left\{\langle \boldsymbol{Z}, \boldsymbol{a}_i\boldsymbol{a}_i^T\rangle\right\}_{i=1}^m, \quad \forall\ \boldsymbol{Z} \in \mathbb{R}^{n\times n}.$$

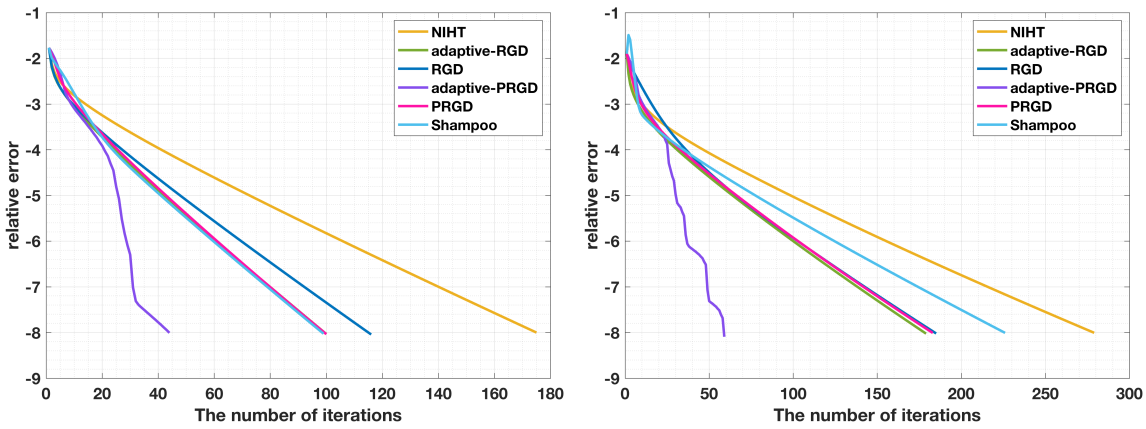
那么，(8) 等价于

$$\mathcal{A}\boldsymbol{X} = \boldsymbol{y},$$

其中 $\boldsymbol{X} = \boldsymbol{x}\boldsymbol{x}^T$。由于 $\boldsymbol{X}$ 与 $\boldsymbol{x}$之间存在一一对应关系，因此我们可以寻求从 $\mathcal{A}\boldsymbol{X} = \boldsymbol{y}$ 中重构秩一半正定矩阵 $\boldsymbol{X}$ ，而不是重构 $\boldsymbol{x}$。因此，相位恢复可以表述为一个低秩矩阵恢复问题（1.1）和（1.2），其中 $n_1 = n_2 = n$， $r = 1$，以及 $\boldsymbol{A}_i = \boldsymbol{a}_i\boldsymbol{a}_i^T$。

我们生成长度为 $n = 128$ 的底层信号 $\boldsymbol{x}$，其元素为独立同分布（i.i.d.）的高斯分布元素，并生成测量矩阵 $\boldsymbol{A}$，其元素同样为独立同分布的高斯分布元素。对于所有实验，我们设定 $m = 6n$。由于秩1矩阵 $\boldsymbol{X}$ 的自由度为 $2n - 1$，因此过采样率约为 $1/3$。对于所有算法，我们均采用停止准则 $\frac{\|\mathcal{A}\boldsymbol{X}_t - \boldsymbol{y}\|}{\|\boldsymbol{y}\|} \le 10^{-8}$ 。

图 8(a) 所示结果表明，与自适应-RGD、RGD 和 PRGD 算法相比，自适应-PRGD 算法所需的迭代次数显著更少。具体而言，自适应-PRGD 的迭代次数约为自适应-RGD 和 RGD 的三分之一，而 PRGD 的迭代次数与自适应-RGD 相当。该实验结果表明，我们的数据驱动度量对相位恢复问题非常有效。然而，在当前实现中，自适应-RGD 的计算时间少于自适应-PRGD，因为自适应-RGD 无需显式计算梯度 $\boldsymbol{G}_t$，而自适应-PRGD 需要显式表达式 $\boldsymbol{G}_t$，这会耗费较多时间。未来的研究将探索高效实现用于相位恢复的 PRGD 的方法。



(a) Gaussian(real) (b) Gaussian(complex)

图 8：模拟数据上相位恢复的迭代次数结果。未知信号的大小为 128。

相位恢复问题及其算法可以从实数情形推广到复数情形，此时未知向量 $\boldsymbol{x}$ 和测量矩阵 $\boldsymbol{A}$ 均为复值。我们针对复数相位恢复进行了实验，并比较了所有算法。结果见图 8(b)。研究结果与图 8(a) 所示实数情形下的结果一致。

5 结论与未来方向

本文针对低秩矩阵恢复问题提出了一种预条件黎曼梯度下降（PRGD）算法。预条件子由测量数据和迭代数据构造，且易于计算。我们证明，在受限等距性质（RIP）成立的条件下，PRGD 能够线性收敛至真实的低秩矩阵。我们评估了 PRGD 在多种低秩矩阵恢复问题上的性能，包括低秩矩阵补全、低秩矩阵感知和相位恢复。实验结果表明，PRGD 优于 RGD，能够减少迭代次数；在矩阵补全任务中，其速度最高可比 RGD 算法快 10 倍，因此是低秩矩阵恢复中更高效的选择。总体而言，我们认为所提出的预条件子和 PRGD 算法对于低秩矩阵恢复具有显著的效率优势。

然而，正如本文第4.3 节所示，对于相位恢复问题，尽管与 RGD 相比，PRGD 可以减少迭代次数，但在当前算法实现中，计算时间尚未进一步缩短。未来，我们将探索解决这一问题的方法。此外，我们提出的新预条件子和 PRGD 算法可扩展至其他恢复问题，例如低秩张量补全。我们还可以针对不同问题构造新的、更有效的预条件子。未来我们计划继续开展这方面的研究。

致谢

Jian-Feng Cai 部分获得香港研究资助局 GRF 16306821 和 GRF 16310620，以及香港创新及科技基金 MHP/009/20 的资助。Fengmiao Bian 也部分获得上海交通大学优秀博士毕业生发展奖学金的资助。

参考文献

[1] D. L. Donoho. 压缩感知。 *IEEE Trans. Inform. Theory*, 52(4):1289–1306, 2006.[2] A. Argyriou, T. Evgeniou and M. Pontil. 多任务特征学习。 *In Advances in Neural Informaiton Processing Systems NIPS* 41 48, 2007 , , 页码 – 。[3] A. Kyrillidis and V. Cevher. 硬阈值方法的矩阵技巧。 *J. Math. Imaging Vision* 48 235 265, , : – 。 2014
[4] A. Lewis and M. Overton. 基于拟牛顿法的非光滑优化。 *Math. Program.*, 141(1-2):135– 163, 2013.[5] B. Recht. 一种更简单的矩阵补全方法。 *J. Mach. Learn. Res.*, 12:3413–3430, 2011.[6] B. Recht, M. Fazel and P.A. Parrilo. 通过核范数最小化求解线性矩阵方程，保证获得最小秩解。 *SIAM Rev.*, 52:471–501, 2010.[7] B. Vandereycken. 通过黎曼优化进行低秩矩阵补全。 *SIAM J. Optim.* 23 2 1214 1236, , ( ): – 2013
。[8] C. Chen, B. He and X. Yuan. 通过交替方向法进行矩阵补全。 *IMA J. Numer. Anal.*, 32 1 227 245, 2012 ( ): – 。[9] C. Mu, Y. Zhang, J. Wright and D. Goldfarb. 可扩展鲁棒矩阵恢复：Frank-Wolfe 与近端方法相结合。 *SIAM J. Sci. Comput.*, 38(5):A3291–A3317, 2016.[10] D. Goldberg, D. Nichols, B. Oki and D. Terry. 利用协同过滤编织信息织锦。*Communications of the ACM*, 35(12):61–70, 1992.[11] D. Gross. 从任意基中的少量系数恢复低秩矩阵。 *IEEE Trans. Inform. Theory*, 57 3 1548 1566, 2011 ( ): – 。[12] D. Kressner, M. Steinlechner and B. Vandereycken. 通过黎曼优化进行低秩张量补全。*BIT. Numerical Mathematics*, 54(2):447–468, 2014.[13] E. Cand ` *Found. Comput. Math.* 9 717 772, es 2009
and B. Recht. 通过凸优化实现精确矩阵补全。 , : – 。

[14] E. Cand`es and T. Tao. 线性规划解码。 *IEEE Trans. Inform. Theory*, 51:4203–4215, 2005.[15] E. Cand
` *IEEE Trans. Inform.* es
Theory 56 5 2053 2080, 2009
and T. Tao. 凸松弛的力量：近乎最优的矩阵补全。 , ( ): – 。[16] E. Cand`es, J. Romberg and T. Tao. 稳健不确定性原理：从高度不完整的频率信息中精确重建信号。 *IEEE Trans. Inform. Theory*, 52(2):489–509, 2006.[17] E. Cand`es, T. Strohmer and V. Voroninski. PhaseLift：通过凸规划从幅值测量中实现精确且稳定的信号恢复。
Comm. Pure Appl. Math., 66(8):1241–1274, 2013.[18] E. Cand
` *IEEE Trans.* es,
Inform. Theory 61 4 2015
X. Li and M. Soltanolkotabi. 通过Wirtinger流进行相位恢复：理论与算法。 , ( ), 。[19] E. Cand
` *Journal of the ACM* 58 3 1 37, es,
2011
X. Li, Y. Ma and J. Wright. 稳健主成分分析？ , ( ): – 。[20] F. Harper and J. Konstan. MovieLens数据集：历史与背景。*Intelligent Systems* 5 19 1 19, 2016 *ACM Transactions on Interactive* ,
( ): – 。[21] G. H. Golub and C. F. Van Loan. *Matrix Computations*。约翰斯·霍普金斯出版社，巴尔的摩，2013 年。[22] H. Cai, J.-F. Cai and K. Wei. 用于稳健主成分分析的加速交替投影。*J. Mach. Learn. Res.*, 20:1–33, 2019.[23] J. Blanchard, J. Tanner and K. Wei. 共轭梯度迭代硬阈值：用于压缩感知和矩阵补全的共轭梯度迭代硬阈值方法。
Inf. Inference, 4(4):289–327, 2015.[24] J. Bolte, S. Sabach and M. Teboulle. 用于非凸非光滑问题的近端交替线性化最小化。 *Math. Program. Ser. A*, 146(1-2):459–494, 2014.[25] J. Duchi, E. Hazan and Y. Singer. 用于在线学习和随机优化的自适应次梯度方法。*J. Mach. Learn. Res.*, 12:2121–2159, 2011.[26] J.-F. Cai and K. Wei. 通过黎曼优化求解无相位方程系统，并实现最优采样复杂度。 *to appear in J. Comput. Math.*, 2022年。[27] J.-F. Cai, E. Cand
` *SIAM J.* es
Optim. 20 4 1956 1982, 2010
and Z. Shen. 用于矩阵补全的奇异值阈值算法。 , ( ): – 。[28] J.-F. Cai, J. Li and D. Xia. 通过黎曼优化实现可证明的张量列格式张量补全。
J. , ( ): –
Mach. Learn. Res. 23 123 1 77, 2022
。[29] J.-F. Cai, T. Wang and K. Wei. 通过低秩汉克尔矩阵补全实现频谱稀疏信号重建的快速且可证明算法。
Appl. Comput. Harmon. Anal., 46(1):94–121, 2019.[30] J. Miao, T. Ishikawa, Q. Shen and T. Earnesty. 扩展X射线晶体学，使其能够对非晶材料、细胞和单蛋白质复合物成像。
Annual Review of Physical Chemistry 59 387 410, , : – 。
2008
[31] J. Nocedal. 以有限存储方式更新拟牛顿矩阵。 *Math. Comp.*, 35(151):773–782, 1980年。[32] J. Tanner and K. Wei. 用于矩阵补全的归一化迭代硬阈值。 *SIAM J. Sci. Comput.*, 35 5 104 125, 2013 ( ):S –S 。[33] J. Tanner and K. Wei. 通过交替最速下降法进行低秩矩阵补全。
Appl. Comput. , : – 。
Harmon. Anal. 40 417 429, 2016
[34] K.-C. Toh and S. Yun. 用于核范数正则化最小二乘问题的加速近端梯度算法。 *Pac. J. Optim.*, 6:615–640, 2010.
[35] K. Wei, J.-F. Cai, T. Chan and S. Leung. 黎曼优化用于低秩矩阵补全的保证。*Inverse Probl. Imaging*, 14(2):233–265, 2020年。[36] K. Wei, J.-F. Cai, T. Chen and S. Leung. 黎曼优化用于低秩矩阵恢复的保证。
SIAM J. Matrix Anal. Appl., 37(3):1198–1222, 2016.[37] M. Kliesch, R. Kueng, J. Eisert and D. Gross. 通过少量测量保证量子过程恢复。*Quantum*, 3:171, 2019.[38] M. Tao and X. Yuan. 从不完整且含噪观测中恢复矩阵的低秩分量和稀疏分量。 *SIAM J. Optim.*, 21(1):57–81, 2011.[39] P. Jain, P. Netrapalli and S. Sanghavi. 使用交替最小化进行低秩矩阵补全。 *In Proceed-*

ings of the 4th Annual ACM Symposium on Theory of Computing, ACM, 第 665–674 页，2013年。[40] P. Jain、R. Meka 和 I. Dhillon。通过奇异值投影实现有保证的秩最小化。 *In Proceedings of the Neural Information Processing Systems Conference* 937 945, 2010 ，第 – 页。

[41] Q. Zheng 和 J. Lafferty。利用 Burer-Monteiro 因子分解和梯度下降进行矩形矩阵补全的收敛性分析。 *arXiv preprint arXiv: 1605.07051*，2016年。[42] R. Harrison。晶体学中的相位问题。 *Journal of the Optical Society of America A* 10 5 1046 1055, , (): –。1993

[43] R. Sun 和 Z.-Q. Luo。通过非凸因子分解实现有保证的矩阵补全。 *IEEE Trans. Inform.* , : –。*Theory* 62 6535 6579, 2016

[44] S. Ma、D. Goldfarb 和 L. Chen。用于矩阵秩最小化的不动点和 Bregman 迭代方法。*Math. Program.*，128(1):321–353，2011年。[45] S. Tu、R. Boczar、M. Simchowitz、M. Soltanolkotabi 和 B. Recht。通过 Procrustes 流求解线性矩阵方程的低秩解。 *In Proceedings of the International Conference on Machine Learning PMLR* 964 973, , 第 – 页。2016

[46] S. Zheng、W. Huang、B. Vandereycken 和 X. Zhang。针对 Hermitian 半正定固定秩约束，使用三种不同度量的黎曼优化：扩展版本。 *arXiv preprint arXiv: 2204.07830*，2022年。[47] T. Tong、C. Ma 和 Y. Chi。通过缩放梯度下降加速病态低秩矩阵估计。*J. Mach. Learn. Res.*，22:1–63，2021年。[48] T. Zhang 和 Y. Yang。通过流形优化实现稳健主成分分析。 *J. Mach. Learn. Res.*，19(1):3101–3139，2018年。[49] V. Gupta、T. Koren 和 Y. Singer。Shampoo：预条件随机张量优化。*Conference on Machine Learning PMLR* 1842 1850, 2018 *In International* ，第 – 页。[50] W. Huang 、 K. Gallivan 和 X. Zhang。通过低秩黎曼优化方法求解 PhaseLift。 *International Conference on Computational Science, ICCS*，80:1125–1134，2016年。[51] W. Huang、K. Gallivan 和 X. Zhang。针对复半正定约束，通过低秩黎曼优化方法求解 PhaseLift。 *SIAM J. Sci. Comput.*，39(5):B840–B859，2017年。[52] X. Yi、D. Park、Y. Chen 和 C. Caramanis。通过梯度下降实现稳健主成分分析的快速算法。 *In Advances in Neural Informaiton Processing Systems NIPS* 4152 4160, 2016 ，第 – 页。

[53] Y. Amit、M. Fink、N. Srebro 和 S. Ullman。揭示多类别分类中的共享结构。 *In Proceedings of the 24th International Conference on Machine Learning ICML* 2007 ，。[54] Y. Chen 和 E. J. Cand`es。求解随机二次方程组几乎与求解线性系统一样简单。 *In Advances in Neural Informaiton Processing Systems, NIPS*，2015年。[55] Y. K. Liu。通过 Pauli 测量实现通用低秩矩阵恢复。*Processing Systems NIPS* 1638 1646, 2011 *In Advances in Neural Information* ，第 – 页。[56] Y. Luo、X. Li 和 A. Zhang。非凸因子分解与流形表述在低秩优化中几乎等价。 *arXiv preprint arXiv:2108.01772*，2021年。[57] Y. Xu、W. Yin、W. Wen 和 Y. Zhang。一种用于具有非负因子的矩阵补全的交替方向算法。 *Front. Math. China*，7(2):365–384，2012年。[58] Z. Lin、M. Chen、L. Wu 和 Y. Ma。用于受损低秩矩阵精确恢复的增广拉格朗日乘子法。 *arXiv preprint arXiv: 1009.5055*，2010年。